

THE BRITISH LIBRARY

BRITISH THESIS SERVICE

TITLE

AN EXPERT SYSTEM FOR X-RAY ROCKING
CURVE ANALYSIS USING ANALOGICAL
REASONING.

AUTHOR

Richard William
HENSON

DEGREE

Ph.D

AWARDING BODY

Warwick University

DATE

1993

THESIS NUMBER

DX182706

THIS THESIS HAS BEEN MICROFILMED EXACTLY AS RECEIVED

The quality of this reproduction is dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction. Some pages may have indistinct print, especially if the original papers were poorly produced or if awarding body sent an inferior copy. If pages are missing, please contact the awarding body which granted the degree.

Previously copyrighted materials (journals articles, published texts etc.) are not filmed.

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no information derived from it may be published without the author's prior written consent.

Reproduction of this thesis, other than as permitted under the United Kingdom Copyright Designs and Patents Act 1988, or under specific agreement with the copyright holder, is prohibited.

AN EXPERT SYSTEM FOR X-RAY ROCKING CURVE ANALYSIS
USING ANALOGICAL REASONING

By

Richard William Henson

Submitted for the degree of Doctor of Philosophy
to the Higher Degrees Committee
University of Warwick

Department of Engineering
University of Warwick, Coventry, U.K.

August 1993

Contents

Contents	I
List of figures	VIII
List of tables	XI
Acknowledgements	XIII
Declaration	XIV
Summary	XV
1 Introduction	1
1.1 Introduction	1
1.2 Organisation of Thesis	1
1.3 Review of Chapter 2	2
1.4 Review of Chapter 3	2
1.5 Review of Chapter 4	3
1.6 Review of Chapter 5	5
1.7 Review of Chapter 6	6
1.8 Review of Chapter 7	7
1.9 Review of Chapter 8	9
2 Current Expert Systems	9
2.1 Introduction	9
2.2 The Architecture	10
2.3 Developments in Artificial Intelligence	12
2.3.1 STRIPS	14
2.3.2 NOAH	18

2.3.3	HEARSAY	20
2.3.4	LS-1	23
2.4	Important Developments in Expert Systems	27
2.4.1	MYCIN - a system for medical diagnosis of Infections	28
2.4.2	INTERNIST - a system for general medical diagnosis	31
2.4.3	DENDRAL - a system for analysing chemical compounds	37
2.4.4	MOLGEN - a system to aid in design of biological experiments	41
2.5	The Role of Expert Systems	44
2.6	Conclusions	45
3	Control without Knowledge	47
3.1	Introduction	47
3.2	Problem Spaces	47
3.2	Classification of Problems	49
3.4	Search Strategy	51
3.5	Direction of Search	54
3.6	Heuristic Search	57
3.6.1	Representation	58
3.6.2	Matching Procedure	59
3.6.3	Conflict Resolution	60
3.7	Types of Heuristic Search (Weak Methods)	60
3.7.1	Generate-and-Test	61
3.7.2	Hill Climbing	62
3.7.3	Best First	64
3.8	The Use of Constraints	66

3.9	Conclusions	67
4	Control with Knowledge	68
4.1	Introduction	68
4.2	Knowledge Representation	68
4.2.1	Logical Representations	69
4.2.2	Semantic Networks	72
4.2.3	Frames	76
4.2.4	Production Rules	81
4.2.5	Knowledge Representation Overview	82
4.3	Methods of Inference	83
4.3.1	Logical Inference	85
4.3.1.1	Non-monotonic Logics	90
4.3.1.2	Automation of Proof	93
4.3.1.3	Resolution	96
4.3.1.4	Reasoning	97
4.3.2	Statistical Reasoning	103
4.3.2.1	Bayesian Logic	105
4.3.2.2	Fuzzy Logic	114
4.3.2.3	Dempster Schafer Calculus	116
4.3.3	Discussion	120
4.4	Conclusions	122
5	A Prototype Expert System Shell for	
	X-Ray Rocking Curve Analysis	123
5.0	Introduction	124
5.1	What is X-Ray Rocking Curve Analysis	124
5.1.1	Diffraction Theories	127
5.1.2	Simulating a Rocking Curve	130

5.1.3	Descriptions of Rocking Curves	133
5.2	Is X-Ray Rocking Curve analysis a suitable domain for modelling	139
5.3	A Cognitive Analysis of the Domain	141
5.3.1	Specification of Inputs and Outputs	142
5.3.2	Specification of Search	142
5.3.3	Specification of Inference	143
5.3.4	Specification of Representations	144
5.3.5	Domain Input and Output Characteristics	145
5.3.6	Domain Inference Characteristics	147
5.3.7	Domain Search Characteristics	149
5.3.8	Domain Representation Characteristics	150
5.4	The Expert System Core	153
5.4.1	Knowledge Base	153
5.4.1.1	Frames	155
5.4.1.2	Production Rule Knowledge	157
5.4.2	Inference Engine	162
5.4.2.1	Frame Inference Engine (The Agenda)	162
5.4.2.2	The Production Rule Engine	166
5.4.2.3	Demon Logic	171
5.5	The Prover	175
5.6	Reasoning Methodology	176
5.7	Database	178
5.8	Questions	180
5.8.1	Help	183
5.8.2	How/Why	183
5.9	System Operation	184
5.10	Conclusions	187

6	Knowledge Elicitation and	
	X-Ray Rocking Curve Analysis	188
6.0	Introduction	188
6.1	Knowledge Elicitation Methods	189
6.1.1	Intra-personal Methods	190
6.1.1.1	Interviews	190
6.1.1.2	Protocol Analysis	193
6.1.1.3	Discussion	195
6.1.2	Abstractive Methods	196
6.1.2.1	Classification	196
6.1.2.2	Multi Dimensional Scaling	197
6.1.2.3	Concept Sorting	198
6.1.2.4	Goal Decomposition	199
6.1.2.5	Machine Induction	199
6.1.2.6	Discussion	201
6.2	The X-Ray Rocking Curve Analysis Interviews	202
6.2.1	General Analysis	204
6.2.2	Detailed Elicitations	205
6.2.3	Discussion	206
6.3	A Experimental Feature Extraction Technique	207
6.3.1	Concept Formation	208
6.3.2	A Knowledge Elicitation Design Using Concept Formation	210
6.3.3	X-Ray Rocking Curve Prototypes	214
6.4	Procedural Knowledge and X-Ray Rocking Curve Prototypes	215
6.5	Method and Results of the Elicitation Process	217
6.5.1	Experimental Framework I	217

6.5.2	Experimental Framework II	220
6.5.3	Analysis of Results	225
6.6	The Impact of Knowledge Elicitation on the Expert System Design	226
6.7	Conclusions	230
7	The Development of Deep Reasoning and Representation in Expert Systems	232
7.0	Introduction	232
7.1	Deep Knowledge	234
7.1.1	What Constitutes Deep Knowledge in Expert Systems?	235
7.1.1.1	Problem Solving	236
7.1.1.2	Interfacing	236
7.1.1.3	System Maintainability	238
7.1.1.4	Knowledge Elicitation	240
7.1.1.5	The Design Process	241
7.1.1.6	Consultation Process	242
7.1.2	Models of Input for a Deep Design Approach	242
7.1.2.1	Learning by Examples	243
7.1.2.2	Memory Structures	244
7.1.2.3	Analogy	245
7.2	An Expert System Model for Deep Reasoning	247
7.2.1	A Re-definition of Problem Solving	248
7.2.2	A Model of a Consultation	249
7.2.3	How Core fits to the proposed Deep Model	252
7.3	The Implementation of Deep Expert System	254
7.3.1	The Conceptual Database	255
7.3.1.1	Representation	256

7.3.1.2	Searching for an Object	258
7.3.2	The Analogical Reasoner	262
7.3.2.1	Control of the Analogical Cycle	263
7.3.2.2	The Control Sequence for Analogical Reasoning	267
7.3.2.3	Probability Assessments of Analogical Reasoning	268
7.3.3	The Analogical Processes	272
7.3.3.1	Target Description	273
7.3.3.2	Source Selection	275
7.3.3.3	Mapping	279
7.3.3.4	Evaluation	281
7.3.3.5	Consolidation	285
7.3.3.5.1	Accomodation	286
7.3.3.5.2	Maturation	288
7.3.3.5.3	Affiliation	291
7.4	The Deep Expert System Shell and the Model	294
7.5	Overview of the X-Ray Rocking Knowledge and Deep Reasoning	295
7.6	Conclusions	298
8	Conclusions	299
8.0	Overview	299
8.1	Overcoming the Bottleneck of Knowledge Based Systems	300
8.2	A Search for Deep Knowledge Representation	300
8.3	Results of an Analogical Framework	301
8.4	Implications of Analogical Framework	302
8.5	Limitations and Future Developments	303

Bibliography	305
References	307

Appendicies	317
1 Interviews with Experts - raw data	317
2 Analysis of Experimental Framework I	378
3 Analysis of Experimental Framework II	384
4 Program Run	388
5 Abbreviations	406

List of Figures

2.1 Typical Expert System Configuration	11
2.2 Blocks World Problem for STRIPS	15
2.3 Example of a Contradiction using CRICITS	20
2.4 The HEARSAY Blackboard for Typical Utterance	21
2.5 Schematic of the LS-1	25
2.6a Cross-over of Parent Search Sequences	27
2.6b Inversion of Parent Search Sequences	27
2.6c Mutation of Parent Search Sequence	27
2.7 Associative Network of INTERNIST Disease Tree	33
2.8 Sample Database Network for INTERNIST	36
2.9 Problem Tree for Unspecified Problem	38
2.10 System Flowchart for DENDRAL	40
3.1 A Problem Space for Typical Problem Domains	48
3.2 A Sample Problem Tree for Unspecified	

	Problem	51
3.3	The Problem of the Sideways Chaining Method	56
3.4	Flow Diagram for Generate-and-Test Heuristic	62
3.5	Flow Diagram for Hill Climbing Heuristic	63
3.6	Flow Diagram for Best Path Heuristic	65
4.1	The Basic Binodal Unit of a Semantic Network	73
4.2	Semantic Network for the Node Deformation	75
4.3	A Frame Outline of Concept Experimental Curve	77
4.4	Object Concept for Frame Experimental Curves	79
4.5	An Instantiated Frame for Experimental Curve	80
4.6	Laws of Re-expression for all Formulae	95
4.7	Inferencing Arc for Interference on Peak	111
4.8	Inferencing Net with Multiple Arcs	113
5.1	The Optics of Double X-Ray Diffraction	125
5.2	A Layered Structure and its Associated X-Ray Rocking Curve	126
5.3	The Demonstration of Bragg's Law for X-Ray Diffraction	128
5.4	Typical Features in a Rocking Curve	133
5.5	Expert System Architecture for X-Ray Rocking Curve Analysis	154
5.6	Nested List Structure of Each Frame	155
5.7	A Typical Frame Structure for Expert System	156
5.8	Three Rules from the Storage Frame for the Production Rule Knowledge Base	158
5.9	The Structure of a Single Production Rule	159
5.10	Part of the Constraint Frame Structure	160

5.11	The Structure of the Agenda or Controller	163
5.12	The Structure of the Data Dictionary	164
5.13	The Control Cycle for the Agenda and Frames	166
5.14	Example of a Backward Chaining Rule-Tree	168
5.15	The Storage Frame for Demon Logic Rules	171
5.16	An Example Structure of a Single Demon Rule	172
5.17	Example Rule Tree Generated by Demon Logic	174
5.18	Sideways Chaining Method for Expert System	177
5.19	Structure of Database Frame	180
5.20	The Overall Database Entry Structure	181
5.21	An Extract from Question Frame for X-Ray Rocking Curve Domain	183
6.1	Distribution of Random Dot Prototypes	210
6.2	Prototype Recall for Expertise	219
6.3	Prototype Categorisation for Expertise	219
6.4	Recall of Each Plot Type Against Feature Type	224
6.5	Prototype Recall Against Expertise	225
7.1	The Analogical Reasoning Process	247
7.2	Overall Expert System Learning Structure	250
7.3	Schematic of a Deep Expert System Architecture	254
7.4	Representation of a Two Dimensional Probability Matrix for Complexity and Peak-Asymmetry	256
7.5	The Conceptual Database Search Process	260
7.6	The Structure of the Deep Data Dictionary	263
7.7	The Analogical Cycle of Expert System Shell	264
7.8	The Control Process of the Agenda	266

7.9	The Production of a Target Prototype from the Expert System Core	274
7.10	The Generation of Entry Point into Conceptual Database from Reduced Feature	276
7.11	Feature Tables for X-Ray Rocking Curve Domain	278
7.12	Venn Diagram of the Mapping Process	280
7.13	Set Definition for Evaluation Function	282
7.14	Search Tree of Conceptual Database	289
7.15	Constraint Graph for Affiliation Process	293

List of Tables

3.1	Table of Problem Solving Characteristics	50
4.1	The Logical Connectives for Logic Systems	70
4.2	The Basic Rules of Logical Inference	86
4.3	The Rules of Combination for Probability Functions	119
5.1	Structural Parameters and their Effect on the Rocking Curve Profile	136
5.2	Suitability of Domain Characteristics for Knowledge-Based Approach	139
5.3	Summary of Techniques used within Expert System	152
6.1	Classification of Prototype Distortions Used in Training	223
7.1	Structural and Procedural Equivalences between the Expert Core and the Deep Model	253

7.2	The Conditional Application of $P(R/T)$ to $P(E)$	271
7.3	Probability of Consolidation given the Evaluation of Target and Source	285

Acknowledgements

I wish to thank Dr. T. Tjahjadi and Professor K. Bowen for their constant encouragement and excellent supervision throughout this project. I would also like to thank the following participants who provided my experimental and research data, without which this thesis would not have been possible: Members of Bebe Scientific Instruments, Durham, especially Dr. N. Loxley and Professor B. Tanner and Staff and students of the Engineering Department.

Declaration

The work described in this thesis was carried out in the Department of Engineering at Warwick University between February 1989 and August 1993.

Throughout the period of this research programme I have not registered for any other award of the University of Warwick, nor with any other degree-awarding body: no material content in this thesis has been used in any submission for an academic award.

Finally, my thanks go to the following people with whom I had many fruitful discussions about Artificial Intelligence: Terry Warwick and Takis Markomichalis.

Summary

This thesis examines the contribution of an adaptive expert system architecture to the field of X-ray Rocking Curve Analysis. The domain of X-ray Rocking Curve Analysis is used as an example to illustrate how a formal computer architecture can be enhanced through the principles of analogical reasoning to provide a deep knowledge of the domain.

A conventional expert system core holds knowledge of a target problem in the form of frames, production rules and confidence factors. Through logical inference and demon logic, a reasoning cycle instantiates the frame structure and creates a new set of classes that represents the solution to the problem. The solved problem is a linked set of data held within the frame structure and complete knowledge across all domains held in a common on-line datastore.

Knowledge elicitation reveals X-ray Rocking Curve Analysis to be a strongly visual task, which cannot be completely encoded within the expert system core. Through the application of the concept formation methodology, a set of key visual features (peak density, peak count, peak type) have been elicited from the X-ray Rocking Curve domain. The key features are used as a probability index for referencing previously solved problems.

Structurally, the expert system core is embedded in a five staged analogical problem solving cycle consisting of: Targetting - building a description of the current problem; Source Selection - selecting a problem from a set of previously solved problems; Mapping - adding additional reasoning from the source to the target; Evaluation - a mathematical evaluation of the closeness of fit between the selected source and target; and Consolidation - the modification of the source based on the results of the evaluation.

The key features provide the link between the target and source problems, providing a practical solution to isomorphic comparisons from inexact mapping. Statistical inference is used to enumerate between problems and allow the analogical inferencing to operate without exhaustive computation. A set of consolidative algorithms have been implemented for modifying the source data. It is these algorithms that give the expert system its adaptive characteristics.

The analogical cycle provides a way of both guiding problem solving, and adding and adapting examples of previous cases. The expert system no longer behaves in the same manner each time it operates, but adapts its solutions by modifying the locatability of source information within a 3D probability array. These locations and the data held within is the deep knowledge of the domain that is achieved as the expert system is used to solve problems. Solutions are thereby not fixed, but evolve.

Chapter 1

Aims and Objectives

1.1 Introduction

This thesis outlines the development of an expert system for X-ray rocking curve analysis. The aim of the research is to extend the capacity of expert systems by using cognitively compatible structures within a formal architecture. The focus of the research will be on knowledge elicitation techniques for extracting deep knowledge from the domain along with the analogical structures that support it. The result of this development is an expert system that has the capacity to adapt to the 'world' in which it operates by modifying its cognitive structures. These structures are general and modular, and can be emptied of knowledge to provide the developer with a shell.

1.2 Organisation of Thesis

This thesis is comprised of eight chapters and four appendices. Each chapter covers a specific area of expert systems. The following three chapters cover the technical aspects of expert system architectures, and the remaining chapters outline the specifications for a deep expert system. Appendices one to three cover the details of elicitation from the domain whilst the remaining appendices list the expert system's operation.

1.3 Review of Chapter 2

In chapter two a selection of the major technical contributions to the field of expert systems are discussed.

The chapter initially focuses on the underlying techniques used in the field of artificial intelligence, starting from the simple application of the stacked planning procedure of STRIPS, to abstract pre-planning in CRITICS as well as more complex multi-modal systems of the HEARSAY projects. The second half of the chapter moves from a general examination of artificial intelligence, to its application in the formal architecture of four expert systems. The areas of search spaces within chemical expert system DENDRAL, planning and abstraction of the experimental designer MOLGEN, MYCIN and its handling of uncertainty and the associated statistical methods, and knowledge representation within INTERNIST, both medical expert systems, are all discussed, each giving pointers towards the development of deep systems. The aim of the chapter is to introduce the reader to important areas for expert system design.

1.4 Review of Chapter 3

Chapter three looks at problem solving without the use of knowledge. These heuristics rely on search techniques for their success, and this is important to expert systems because there are times when knowledge runs out even though it may still be possible to solve a problem. Initially, the chapter looks at the concepts of problem spaces and the classification of problems. Understanding problem types is necessary because there are no general problem solving algorithms that are suitable for all types of problem. The strategy and direction of search is examined in terms of both its compositional nature and objectives. This gives rise to the concepts of

problem steps or stages and both forward and backward chaining respectively. These are very general techniques of search and frequently used in expert systems. For example, chaining of both types can be used as a mechanism to support an inference engine, and the problem steps or stages formulation can be employed as the task controller behind agenda control mechanisms.

Finally, chapter three outlines the heuristic techniques of search. These are specific algorithms that demand a formal representation of a problem in order to provide a solution. Again, expert system architectures employ these techniques, and indeed so too does the software of this thesis. A reverse hill climbing strategy based on 'centres of gravity' is used to search for problem matches in a database within the expert system for X-ray rocking curve analysis (see chapter seven).

1.5 Review of Chapter 4

Chapter four examines problem solving with knowledge. In its structure, it adopts the expert system convention of dividing the solution of a problem into a domain independent inference engine, and a problem specific knowledge representation. The first part of the chapter is concerned with the various ways of representing a problem. Four main methods are reviewed: logic, semantic nets, frames, and production rules, as each of these has benefits for encoding different types of knowledge, be it declarative, procedural, conditional, evidential or strategic. The possibility of a mixed representation is discussed and this is explored in greater detail at a later stage in the thesis (see chapter 5). The second half of the

chapter focuses on inferencing techniques. Two broad approaches are examined; statistical inference and logical inference. There are a large range of statistical methods, but the three most important are Bayesian logic, Dempster Shafer calculus and Fuzzy or Possibility logic. These techniques are considered important because they capture uncertainty. They allow a knowledge engineer to ascribe importance to events in a way that is not possible using logical inference. All three techniques are reviewed, but a selection of the best for use in the expert system for X-ray rocking curve analysis postponed until later in the thesis.

The logical approach to reasoning is examined in the form of both propositional and predicate logic. These formalisms capture knowledge in representations that can be solved mathematically using approaches such as algebraic inference and resolution. This allows the system to specify the truth of a statement. The extent to which these methods both provide a representation and a inferencing structure is discussed. The programming language PROLOG is an example of this integrated strategy to problem solving. Extensions to the these types of logic are briefly discussed, including default logic, modal logic and autoepistemic logic. Aspects of these formal proof methods can be included within representations. For example, inheritance and defaults are included as part of the frame representations. The cross links of representation and inference are important in the design of an expert system and are discussed in chapter 6.

1.6 Review of Chapter 5

Chapter five is concerned with the design and implementation of a expert system using existing tools and techniques. The chapter is divided into two parts. Part one examines the domain of X-ray rocking curve analysis. It describes the domain in general terms and summarise its main features. Existing artificial techniques outlined in the previous three chapters are identified, categorised and then matched to the requirements of the domain as a set of methods. The problem structures used within an expert system architecture are listed and then associated with a method (see Table 5.4). The aim of this analysis is to produce a high level definition for an expert system core.

Part two of the chapter describes the design and then the implementation of an expert system core. It consists of a mixed representation of frames and production rules, and is controlled using an agenda. The operation of the system is based on a overall problem definition which is initially captured on an agenda as a set of tasks, and from these are generated sets of sub-tasks. The aim of the core is to instantiate the declarative structure of the domain, stored in a frame based model, and use a production rule system to solve individual tasks fired from the slots within each frame. Backward chaining is used to drive the production rules and forward chaining the frame model. Demons are also employed as 'watch-dogs' over the modelling process.

The chapter concludes by stating that the core can be used to solve X-ray rocking curve problems, but is restricted by the capacity of the modeller to capture all knowledge in a single

set of production rules and frames. In trying to capture all knowledge in a single model, the system sacrifices efficiency at the expense of generality. In terms of depth, the system is shallow in its operation because it does not encode the domain's essential visual nature.

1.7 Review of Chapter 6

Knowledge elicitation is introduced as an important topic in chapter six. The chapter is divided into three parts. Part one looks at conventional elicitation technique, part two outlines the conventional elicitation techniques as applied to X-ray rocking curve analysis, and part three expresses a new method of elicitation for visual or iconic knowledge.

In part one knowledge elicitation is divided into two categories, Intra-personal techniques and abstractive techniques. Both models are discussed in terms of there advantages and disadvantages to the domain. In terms of the X-ray domain, only the intra-personal model is used, the reason being that a domain specific abstractive technique is developed in the second half of the chapter. Part two details the intra-personal techniques used on the domain to extract knowledge for the core of the expert system. Part three outlines a formal elicitation technique for X-ray rocking curve analysis. The technique is based on concept formation which was first used by Posner and Keele when studying the formation of ideas from visual images. The elicitation technique is an adaption of this method with a set number of assumptions all of which are described in the chapter. Unlike other elicitation strategies, this technique has an

experimental method and a statistical basis to its verification. The aim of the method is to extract key features from the domain. The importance of these features are that they are experimentally verified and, therefore, constitute 'deep' knowledge of the domain. The problem remains as to how to encode this deep knowledge, and this is the purpose of the next chapter.

1.8 Review of Chapter 7

Chapter seven examines the contribution of analogical reasoning to the encoding of deep knowledge within the expert system environment. The deep knowledge extracted using the experimental elicitation procedure is encoded into an outer analogical cycle that matches the current consultation of the expert system core to a previously encoded set of conceptual structures.

The chapter begins by briefly outlining second generation characteristics for expert system behaviour which are flexible problem solving, complex user interfaces, and good system maintainability. The key to second generation architectures is their openness, so that rather than being shells for knowledge, they are environments for developing knowledge based systems.

Later in the chapter three schemes of deep knowledge, learning by example, memory models and analogical reasoning, are examined. These schemes suggest the nature of deep knowledge. A continuous model of analogical reasoning (targeting, source selection, mapping, evaluation and consolidation) is selected as an appropriate deep structure. A model for a deep

consultation is developed from the continuous analogical model and fitted to the expert system core. Two consultative cycles are developed from the deep consultative model, a short term consultation captured within the expert system core, and a long term adaptive cycle based on the continuous analogical model. As the short term cycle is linked via its data structures to the long term cycle, no one consultation is necessarily the same as the next. Common solutions converge to a single mapping between a current problem definition and a previously resolved solution. Uncommon solutions diverge into separate mappings.

The rest of the chapter deals with the implementation of the continuous model and the equations and definitions used. Four components are identified within the deep architecture, the expert system core that builds a target description (see Chapter 5), conceptual search space that encodes a probability profile of all key features of the problem domain with an associated data store of past solutions, and an analogical reasoner that controls the development of the long term cycle. The chapter ends by summarising the deep model (see Figure 7.14) and giving an overview of deep knowledge and X-ray rocking curve analysis.

1.9 Review of Chapter 8

Chapter 8 is the conclusion of the thesis, and examines the contribution of deep knowledge to the example domain, possible improvements to the architecture, and future directions for this research.

Chapter 2

Current Expert Systems

2.1 Introduction

Expert system technology is the result of the practical implementation of research conducted mainly in the field of artificial intelligence (A.I.). Since the 1960's, the A.I. community has been interested in modelling human cognitive performance, and has been particularly concerned with the fields of general problem solving (Earnst and Newell 1969; Smith 1983), visual perception (Guzman 1967; Clowes 1971; Lowe 1987), language understanding (Erman, Hayes-Roth Lesser and Reddy 1980), and learning paradigms (Mickalski 1983). The MIT robot project was the centre of much of this early research, but failed to produce generalisable results (Dreyfus 1968). This was reflected by the problems encountered when trying to move from the artificial "blocks worlds" to the real world (Michie 1971). What such research did reveal, however, was the complexity of even the simplest human tasks. Due to the domain dependant problem solving requirements of expert systems, research in this area has tended to be less diverse, not centring on the production of "world" solutions, instead focusing on modelling specialist knowledge, covering the topics of knowledge elicitation (Christine and Izak 1991), knowledge representation (Bobrow and Collins 1975), inferencing (Winograd 1980), and user interfaces (Simmons 1986). Some expert systems attempt to duplicate human performance, whilst others model the cognitive processes, and by default, duplicate human

performance. The extent to which an expert system attempts to model human cognition will depend on the approach taken by the researcher. At one end of the spectrum will lie a series of heuristic devices that may or may not reflect cognitive processes, relying on the external manipulations of statistical or mathematical algorithms (Reggia, Nau and Wang 1984), and at the other end of the spectrum will lie systems that attempt to model cognitive processes and, thereby, reflect human performance (Keravnou and Washbrook 1989). In the case of these later systems, the duplication of human performance may or may not be regarded as advantageous since we are generally very good at qualitative judgements, but not very good at repeatability or quantitative judgements (Murrell 1976).

2.2 The Architecture

Part of the reason for the development of expert system technology was the result of a reaction against the failure of early A.I. systems to invent general problem solving routines. It was found that most A.I. systems were constrained to the theoretical worlds they were developed within, and not generalisable to real world problems. It was also discovered that the encoding of domain specific knowledge was one way of improving the performance of A.I. systems, but at the expense of generality. Expert system technology grew out of this impasse, and there developed an overall system architecture that was different from the conventional computer architecture, and comprised of four main elements:

- a) Knowledge base - containing the domain knowledge.
- b) User Interface - translator to/from user/system.
- c) Inference Engine - control mechanism.
- d) Acquisition Mod. - machine learning strategy.

The diagram in Figure 2.1 shows a typical configuration for these four elements which is more or less followed by all expert systems (Hayes-Roth Waterman and Lenant 1983).

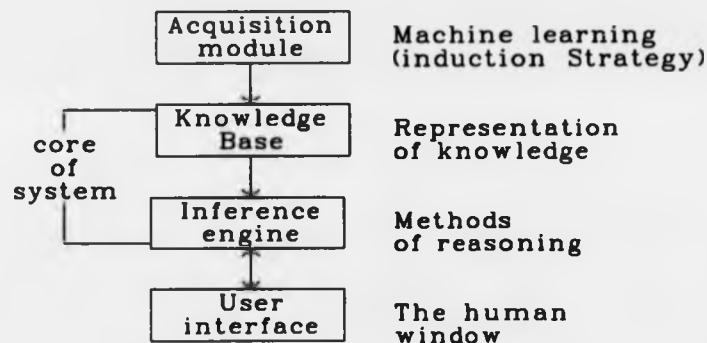


Figure 2.1 Typical Expert System Configuration

With reference to Figure 2.1, the induction module involves developing learning strategies for eliciting knowledge from the user, and there have been a number of attempts to create machine learning (Boose 1986; Dietterich and Michalski 1981). The knowledge base stores this information in specific styles of representation, usually in the form of

one or more of four types (predicate calculus, production rules, frames, semantic nets). The inferencing methods utilise the knowledge in the system to produce a control strategy. In general, a problem is configured as a search space and two types of search strategy (backward chaining, forward chaining) applied when trying to solve it. Backward chaining operates from goal to hypothesis, trying to work out what information is needed to satisfy a particular goal, and forward chaining is data driven, trying to match the data to a hypothesis. The user interface is the final component of the expert system, and involves the way the reasoning of the system is explained to the user i.e. the system justifying its conclusions, and the way the system reacts to the responses of the system i.e. perhaps fitting the questioning strategy to the experience of user.

2.3 Developments in A.I.

Some of the earliest work in the field of A.I. was carried out as part of the MIT robot project. Co-ordinated research into a number of areas of A.I. aimed to build machines capable of behaving in a constrained theoretical world of blocks similar to way in which a human might behave in their environments. Particular aspects of perception were investigated including the transformation of grammar into a syntactic structure and then a semantic format, the visual perception of toy blocks, and the solving of problems in translating the domain into the required format. Simple primitive objects such as blocks and pyramids of different

colour were used in the theoretical worlds and these served as indices for the investigation of object composition and learning paradigms (Winston 1975). The overall aim of this and other similar projects was to build a system that first understood instructions and questions, identified the agents within the theoretical world and then planned the necessary actions to transform the world state into the goal state, upgrading responses to external interactions by modifying operations based on training examples. These systems presented a standard procedure for investigating human performance and highlighted a number of issues, including the need to develop adequate descriptions of domain specific knowledge before attempting practical solutions to specific problems, the critical role played by planning procedures to organise knowledge within the system, the need to define individual problem domains in terms of a restricted search space, and the critical role played by categorisation and classification for representing knowledge. A number of important systems contributing to these issues were developed including:

a) STRIPS - a system for solving multiple goals (Fikes and Nilsson 1971)

b) NOAH - a system for decomposing planning (Sacerdoti 1975)

c) HEARSAY III - a system for multi-level analysis (Balzer, Erman, London, and Williams 1980)

d) LSI - a system for learning (Backman 1990)

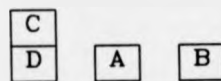
2.3.1 STRIPS

This was a system designed to solve multiple goal situations that interact ie. have dependant states. It used the blocks world as the domain of understanding, and a set of operator to manipulate a goal stack. Three stack operators were used to manipulate the stack ADD, DELETE, and PRECONDITION, and a set of descriptive operators were used to describe the blocks world and the legal actions within it. The descriptive operators were the predicates: ON(x,y), ONTABLE(x), HOLDING(x), CLEAR(x), ARMEMPTY, and actions were planned using a further four predicates: STACK(x,y), UNSTACK(x,y), PICKUP(x), PUTDOWN(x). The world consisted of a series of labelled blocks, a table, and an arm for manipulating each block one at a time, and the aim of the system was to represent the world in a computer, identify the state of the world (the start state), identify a new representation of the world (the goal state) ie the same blocks, but in a different configuration, and plan a sequence of actions using the operators available to interconnected the two via intermediate states. The goal stack functioned from the top downwards and the goals at the top of the stack were identified first, expanded if necessary and then satisfied before moving sequentially onto the next goal in the stack. The system stored the results of each satisfied goal from the goal stack in a database, and that goal was then removed from the stack. The system continued to sequentially operate on each goal in the stack

until all were satisfied, the results being sequentially stored in a database and correspondingly removed from the stack. An empty stack indicated that the system had planned a sequence of operations on the blocks, transforming the start state into the goal state.

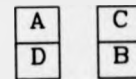
Figure 2.2 shows a simple blocks world problem of four elements, including a description of the start state and the goal state using a limited set of predicates provided by the system.

The states are represented as predicates with each being composed of five logical propositions formed into a compound expression using the AND logical connective.



start state

ON(C,D) ^
 ONTABLE(D) ^
 ONTABLE(A) ^
 ONTABLE(B) ^
 ARMEMPTY



goal state

ON(A,D) ^
 ON(C,B) ^
 ONTABLE(D) ^
 ONTABLE(B) ^
 ARMEMPTY

Figure 2.2 Blocks World Problem for STRIPS to Solve

The first four propositions describe the blocks world and the ARMEMPTY condition is a proposition that states that the system must not be holding an object in either state. Having described the start and goal states of the problem the system sub-divides the problem into its individual propositions. By sub-dividing the problem in this manner Figure 2.2 shows that sub-goals ONTABLE(B) and ONTABLE(D) are both satisfied and, therefore, eliminated from the stack. However, subgoals "C" and "A" are not satisfied and, consequently added to the stack. By inserting each of these unsatisfied goal states into the goal stack, STRIPS defines a problem to be solved by the system and is then able to describe what conditions constitute the current state of "A" and "C" in terms of their relationship to other blocks in the world. The goal stack is, therefore, activated and described thus:

```
ON(A,D)
ON(C,B)
ON(C,B) ^ ON(A,D) ^ ONTABLE(D) ^ ONTABLE(B)
```

In this particular goal stack, ON(A,D) and ON(C,B) are both untrue, and, therefore, the next step for STRIPS is to transform the start state by trying each of the predicates on the top goal to see if the goal state can be matched. These trails continue until such time the stack is empty. STRIPS shows how predicate logic can be used to plan operations on decompositional goals, and works effectively in giving solutions to simple problem provided that the

necessary operators are defined, and the problem can be divided into sub-problems. In the STRIPS system the location of each block is considered an individual problem, but by linking these via a goal stack it is possible to deal with interacting goals. In the example illustrated, the stacking of "C" on "B" is resolved through the interaction of the separate goal to CLEAR(D), the clearing of "D" being a sub-goal of necessary conditions to satisfy STACK(A,D). However, there are two problems with this approach. Firstly, by using predicate logic as the basis for representing information in the world, abstraction of concepts is restricted by the logical constraints of this form of knowledge representation. Secondly, whilst STRIPS highlighted the importance of planning before action, it failed to address the full implications of planning strategy. In this situation, plans are not always formed into separate parts that can be resolved independently in a linear fashion, and this is particularly so as a situation becomes more complex. The skills of an expert are one such example, and it is often the case that expertise is used in the absence of information, and under these circumstances it may be necessary to formulate only partial plans before proceeding to a final solution. STRIPS makes no attempt to partially complete a problem, and only proceeds to the next goal(s) once the problem at the top of the stack has been solved. The planning principles of STRIPS are, therefore, insufficient for providing solutions to complex problems.

2.3.2 NOAH

Nilsson has pointed out that the techniques of STRIPS can be modified using backward chaining from goal to start state, pruning the search tree generated to provide a reasonably sized search space to solve non-linear planning problems (Nilsson 1980). STRIPS tended to work from the start state towards the goal, and, therefore, tried to prove the preconditions before applying the solution. A stack was used to organise the processing of the goals and sub-goals which, whilst adequate for some problems was ineffective for others.

By introducing an alternative method of 'sets of goals' rather than 'stacks of goals' it is feasible to choose from a number of possible goals rather than the top one. However, in order that the selection process is logical, it is necessary to introduce a hierarchy to the goal organisation to differentiate between important goals and inconsequential goals. It is also important to introduce a tree pruning algorithm to reduce the number of operators that might apply to the 'set of goals'. This was previously unnecessary because the stack sequentially restricted access to the top goal only.

The use of abstraction could be used to deal with increased complexity, thereby building operators into larger commands that only evolve solutions at an initially high level. For example, a high level command might be $TOWER(A,B,C)$, which broken down into sub-operators equals $ONTABLE(C) \wedge ON(B,C) \wedge ON(A,B)$. Knowing the moves and preconditions of these three objects (A,B,C) could permit a

quick check of the necessary object states. For example, an overall plan might be to build several TOWERS and implement a search strategy geared to maximising the availability of CLEAR(x) to enable an effective building program. An alternative strategy would be to apply a priority value to solving each goal, and a threshold above which search would be conducted. Plans would be evolved above the threshold, but none would be evolved below such a threshold. This is the approach taken by ABSTRIPS, an extension of STRIPS (Sacerdoti 1974).

NOAH uses similar strategies to those previously outlined, and employs a programming routine called CRITICS to observe the plans produced by the system. CRITICS is used to resolve conflicts when more than one alternative from the set of goals is available. CRITICS works by placing constraints on plans, and in terms of the blocks world, highlights operators whose pre-conditions might undo previously approved operations. For example, if an operation requires a stack of three objects "A", "B", and "C" then CRITICS will order the operations such that contradictions arise if STACK(A,B) occurs before STACK(B,C) when the world goal is (ON(A,B) ^ ON(B,C) ^ ONTABLE(C)), and only one block can be moved at a time. This is shown in Figure 2.3.

CRITICS is also used to eliminate redundant pre-conditions. This overall approach has been labelled the least-commitment strategy, and provides a useful insight into the complexities of planning changes in state within the simple blocks domain.

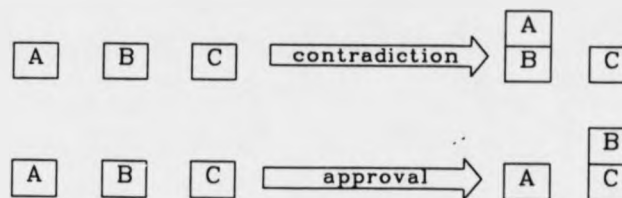


Figure 2.3 Example of a Contradiction using CRITICS

2.3.3 HEARSAY

Both STRIPS and NOAH deal with the planning of operations and the sequences in which they should occur. However, in complex domains what is more central to the operation of the system is the planning of the object relations themselves. The HEARSAY system is a language understanding model that attempts to do exactly this, building interfaces across many sub-domains using a common method of communication. HEARSAY was a system designed to correctly interpret spoken English sentences using established linguistic theories (Hendrix, Sacerdoti, Sagalowicz, and Slocum 1978; Bruce 1975). The particular issue here was the hierarchy of language understanding including the phonetic structure of the sounds, the syntactic structure of the sentence and the semantic structure, where separate problems required different methods of planning. To solve this problem HEARSAY employed a blackboard method of problem solving, which operated on an ascending modular basis. Each module produced results that could be used by other modules higher up the chain, and the results produced by the next module were used

by other modules next in sequence. The results would be written on the blackboard and these could then be understood by other modules as evidence for solving their own problems. The blackboard was constructed along two axes: the Y-axis was the level of analysis dealing with hypotheses about phonemes to complete sentences, and the X-axis the measure of time over which the utterance was measured. Figure 2.4 shows a typical utterance and its layout on the blackboard:

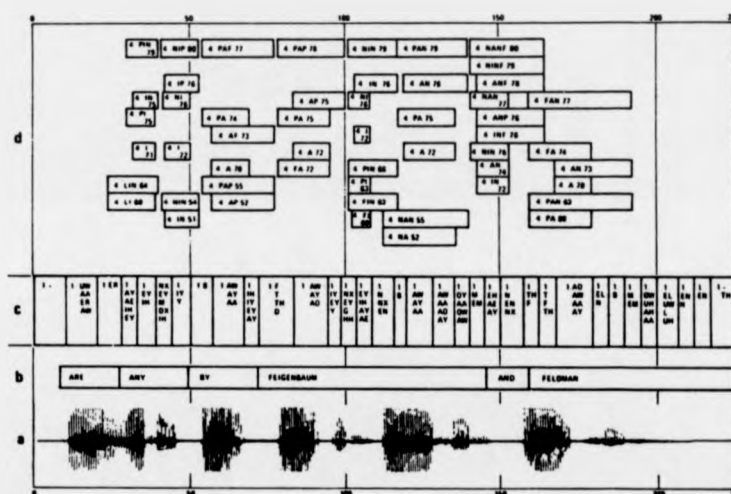


Figure 2.4 The HEARSAY Blackboard for Typical Utterance.

As shown in Figure 2.4 the level of analysis is divided into eight levels:

- a) The waveform of the utterance
- b) The actual words of the sentence
- c) The sound segments

- d) The syllable classes
- e) Each individual word K.S.
- f) Second word K.S.
- g) Word sequences
- h) Phrases

The operation of the blackboard was performed by a series of demons, one for each of the eight levels, and these demons were used to specify which knowledge set to activate given the current evidence available (what is written on the blackboard). When information from the various algorithms, one set of algorithms for each level, was placed on the blackboard, one or more demons had their conditions satisfied. The activated demons then prompted their knowledge set into action, requesting that the blackboard be analysed. If more than one demon was activated at a time then a scheduler was used to conduct the search process in an orderly manner. The scheduler made decisions based on ratings produced by each activated K.S. and this provided comparative value that could be used by the scheduler to decide which demon should analyse the blackboard next. Roth-Hayes and Lesser (1977) provide details of the scheduler operation.

HEARSAY's approach to problem solving differs markedly from NOAH and STRIPS, especially as it is trying to solve a real world problem. The contribution of this strategy to problem solving can be summarised as:

a) Providing a method of dealing with multi-levelled information, some at a low level such as phonetic sounds, and some at a higher level such as syntactic structure.

b) Suggesting a sophisticated control structure using demon logic, each demon dedicated to recognising evidence relevant to its domain of control.

c) Organising knowledge into sets, which is both well understood mathematically, and an effective way of structuring knowledge in a 'cognitive' fashion.

2.3.4 LS-I

An important aspect of A.I. applications is the development of learning paradigms. This process ensures that the knowledge base of a system is never static and always capable of being re-defined. An expert's knowledge is also never static and responsive to the situation, indicating that the use of learning strategy is an important issue in expert system design. Learning strategies tend to be of three types, and all can be regarded as problem solving strategies applicable to novel rather than encoded situations:

- a) Learning by analogy
- b) Learning by generalisation
- c) Learning by discovery

Analogical problem solving refers to the mapping together of elements in a source domain and a target domain. The source domain refers to the analogical source from which schematic representations can be formulated for the mapping of the target, and the target domain refers to the problem state the requires resolution.

What is of interest here is that experimental evidence shows that human problem solving tends to use systematic analogies that map to higher order relations ("suggests" or "caused-by") in a one-to-one correspondence from source to target (Lakoff and Johnson 1980).

Learning by generalisation refers to the principle that learning may be result of the allocation of concepts to common memory arrays, and that generalisations are the result of firstly, assigning concepts to the common array and secondly, drawing on the common characteristics of members to make generalisations through common association (Winston 1975:157).

Learning by discovery is different again, and relies on general problem solving techniques. By discovery, one means that an entity acquires knowledge that the user or the designer of the system does not have, and in this sense does not rely on teaching examples to acquire knowledge. Such learning paradigms are especially useful in mathematical domains where knowledge is monotonic and simple to control (Vere 1975).

The system illustrated here, LS-1, relies on none of the above strategies, but introduces a fourth method, that of a genetic search. The LS-1 system was designed as a prototype

learning system for acquiring specific sets of heuristics, represented as a production rule system, to govern the application of operator sets to solve domain specific problems, and the genetic algorithm was used as a method of search for improving the performance of the system. There are three functional components of LS-1:

a) A problem solving device - an inference engine for applying alternative sets of controls heuristics of the domain task.

b) The critic - an evaluation routine for assessing the success of a given set of control heuristics, and judging performance.

c) A learning device - a genetic searching strategy for generating new heuristics in response to performance.

Figure 2.5 shows how these components fit together to maintain a knowledge base of m structures, each a candidate set of control heuristics to solve the domain problem.

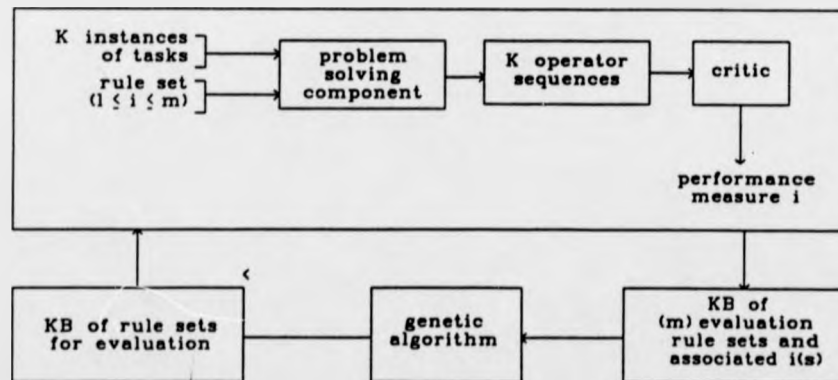


Figure 2.5 Schematic of the LS-1

The tasks cyclically loop through the system, refining performance, and a current hypothesis is generated based on the results of the problem solver. This process is measured by the CRITIC which analyses K operator sequences generated by the rule set. The performance measure is then used by the genetic algorithm to generate off-spring by crossing, mutating, or inverting the most promising rule sets to produce new hybridised rule sets. The hybrids are entered into the knowledge base of the system and re-used by the problem solver to produce a new operator sequence for evaluation.

The success of the genetic algorithm is dependant on producing a rule set with high granularity. In other words, the knowledge representation technique must be capable of reducing the size of the sub-elements in the rule set to a level that allows operator sequences to be crossed without corrupting the knowledge. For example, in Figure 2.6a the cross-over operator is shown producing new off-spring from two high performance rule sets, Figure 2.6b shows the inversion operator transforming a single high performance rule set, and Figure 2.6c shows the mutation operator which operates in background and occasionally introduces or substitutes a random rule to ensure that the search process never reaches a local maxima.

In all figures α must be small to ensure that the break points in the parent rules occur at the operator boundaries. The larger α the greater the probability of breaking within the operator.

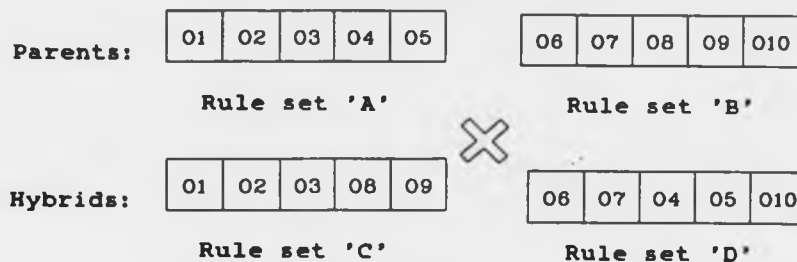


Figure 2.6a Cross-over of Parent Search Sequences

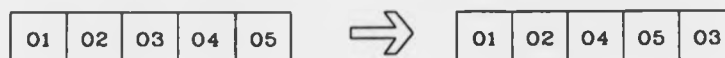


Figure 2.6b Inversion of Parent Search Sequences



Figure 2.6c Mutation of Parent Search Sequence

2.4 Important Developments in Expert Systems

A number of successful expert systems have been developed, and MYCIN, INTERNIST, DENDRAL, and MOLGEN are examples of such systems, with each respectively addressing the management of uncertainty through the application of Bayesian type logics to medical diagnosis, the use of associative network representations again for medical diagnosis, the generate-and-test method for reducing large search spaces in the identification of chemical structures, and the use of abstraction to handle large open ended problems of design for advising on molecular genetic experiments.

2.4.1 MYCIN - a system for medical diagnosis of infections (Shortcliff 1976).

MYCIN is an expert system that has a strongly organised knowledge base for the identification of infectious diseases. The systems knowledge is usually incomplete since the diagnosis of disease is a complex multidimensional domain with conflicting as well as conciliatory symptoms. To express these vagaries MYCIN uses certainty factors (CF) as a measure of belief in the diagnosis of disease, which is influenced by the conditional probabilities of Bayes' theorem.

The control structure of MYCIN is based on a simple production rule system with four components:

- a) Facts
- b) Production Rules
- c) Inference Engine
- d) Heuristics for assessing uncertainty

The facts are stored as triples in the form of CONTEXT-PARAMETER-VALUE: CF. The CONTEXT is an entity expressing some element in the domain, the PARAMETER is an attribute of that entity, and the VALUE is an instance of that PARAMETER. Attached to each triple is a certainty factor and this holds a value between +/-1 as a likelihood measure of the given fact. For example, a 30 year old patient could be expressed as:

PATIENT-AGE-30: .99

In this example the context is PATIENT which has a parameter AGE that has the value 30. One cannot be certain of the patients age without a birth certificate hence .99.

A production rule system is employed to represent the knowledge and both forward and backward chaining are used by the inference engine. The structure of the rules are in the form of an antecedent and a consequent with a CF attached to indicate the certainty of the rule when applied to a problem.

IF antecedent THEN consequent (CF).

The rules are applied in either reasoning direction until the goal triple(s) is instantiated, and this may require questioning the user or utilising the domain knowledge or both.

The inference engine mostly uses backward chaining to reach the goal, deciding which triple to fire next. The control knowledge for this decision is store in a context tree, with the root of the tree forming the starting point from which a hierarchy of templates or rule groups are formed during a consultation, with only those rules in the 'rule group' being considered during the reasoning process.

The Heuristics are the final component of MYCIN and are concerned with calculating the CF of those triples added to the database, the database being a temporary working memory for the system during a specific consultation. The CF is initially calculated as a minimum of the premise for a rule,

consequently the most unlikely event determines the antecedents certainty. This is then multiplied by the CF of the consequent or action part of the rule to given a certainty factor CR. If the triple is new then CR is the new CF, if not then the original CF of the triple becomes CI and a new CF calculated as follows:

CF = CR if triple not in database otherwise:

$$CF = CI + CR(1-CI) \quad CR, CI > 0$$

$$CF = -(|CI| + |CR|(1-|CI|)) \quad CR, CI < 0$$

$$CF = \frac{CI + CR}{1 - \min(|CI| + |CR|)} \quad CI, CR < 0$$

The problem with CFs is that they can only place hypotheses into groups of 'most probable' and 'least probable' diagnosis, and the system has no real mechanism for selecting the best candidate solution. These doubts have been raised by Bachanan et al, and they suggest that the maximal CF value is not necessarily the best hypothesis (Shortliffe, Buchanan, and Feigenbaum 1979). Further to this, there is also a tendency for the CFs to converge too rapidly to one, irrespective of how small the individual CFs of each rule are. So by successively applying rules with a small probability, a 'most probable' diagnosis may be

generated from large amounts of weak evidence. One solution to this may be to apply dampening factors to the probability heuristics, but so far this has not been implemented in MYCIN.

2.4.2 INTERNIST - a system for general medical diagnosis (Miller, Pople, and Myers 1982).

The approach taken by the INTERNIST project differed from that of MYCIN in that it attempted to diagnose a medical condition using the same reasoning strategies as medical experts, making use of the inherent causal relations between the symptoms and the diseases that manifest them. The aim of the system was to identify sets of diseases as candidate hypotheses for explaining the symptoms of the patient, and use a selection strategy to choose between them. This was an attempt at automating the decision-making techniques of the medical practitioner, allowing the system to follow the disease during its various stages in development. By contrast, MYCIN would simply register an increasing CF value for a particular disease entity rather than framing the diagnosis based on the visible symptoms.

The general approach of INTERNIST was to model human cognitive processes by specifying two stages in processing, linking the manifestations of the disease (inflamed liver, vomiting, anaemia) to specific diseases (hepatitis, cirrhosis, metastases):

a) Framing the diagnosis, choosing between a set of mutually exclusive hypotheses.

b) The application of strategy for recalling the diagnosis by identifying the disease that accounts for most symptoms.

The importance of the INTERNIST approach was that the expertise of the clinician was classified as a method for formulating diagnostic tasks, using those tasks to guide additional data gathering. The project highlighted how the clinician tended to set-up a programme of investigation very early on in the consultation even though the probability of the tasks being correct was very low. It was felt that in the design of the system, a focus of reasoning gave it clear guidance in the formulation of hypotheses despite the probable inaccuracy of the approach taken.

An associative network was used to structure the knowledge of INTERNIST, and this consisted of a hierarchical relationship between classified elements in a disease tree. The FORM_OF relation was used to link the elements together and an example tree is shown in Figure 2.7.

The top label All-Diseases inherits all the signs, symptoms and test results exhibited by the patient, whilst further down the tree the inheritance of these characteristics is reduced to the classification set, and eventually onto a specific set of exclusive manifestations exhibited by the patient, the proviso being that the disease has already been classified in the system.

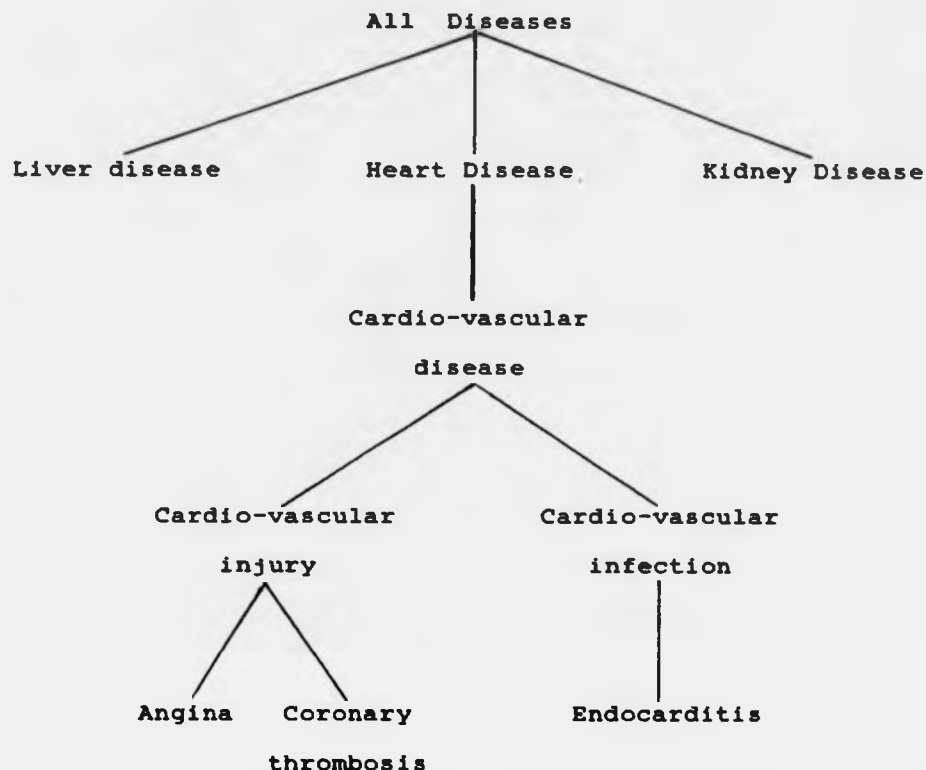


Figure 2.7 Associative Network of INTERNIST Disease Tree

A set of relations is used to link the tree together, and these act as the control mechanism for INTERNIST. Five relations are used in the system as follows:

a) EVOKES

This links the signs and symptoms to the diseases that exhibit these characteristics, and the strength of the association between each element is given a number between 0 and 5, with 0 indicating no EVOKing strength with the

manifestations ruling out the disease, and 5 indicating the highest EVOKing strength with the manifestations suggesting the disease.

b) MANIFESTS

This is the inverse of EVOKES and leads from the disease to a particular set of symptoms, signs and test results. A MANIFESTing strength is used to indicate the strength of association, with 0 suggesting no frequency between the disease and its manifestations, and 5 indicating that the manifestations are always present when the patient has a specific disease.

c) TYPE

This relation is used in the selection of questions to be asked, with priority being given to the least expensive interactions first ie in terms of both financial cost (money and resources) and risks (endangering the patient). The higher the value (0-5) the greater the expense and lower the priority for pursuing this line of questioning.

d) RULEOUT

This relation is used when there are many candidate hypotheses (diseases) that could be used to explain the manifestations. This relation prompts the TYPE relation to take a strong line in questioning to reduce the size of the candidate list of hypotheses.

e) DISCRIMINATE

This relation is a subtle version of RULEOUT, and is used when there are only two or three candidate hypotheses, prompting the TYPE relation to use a fine line of questioning to discriminate between closely associated diseases.

In the operation of these relations, a semantic network for each specific diagnosis is set-up, and stored separately from the knowledge base, in a database for the duration of the consultation. The various manifestation strengths (Mx) and the evoking strengths (Ex) are calculated for each link and used to join the diseases to the manifestations. A TYPE assignment is also given to each disease node to indicate the least expensive line(s) of questioning during the diagnosis. Figure 2.8 illustrates the use of the network for a small sample of heart diseases.

In operating INTERNIST, the process begins by entering the clinical symptoms of the patient into the system. These manifestations generate an initial disease model, consisting of all high level nodes (heart disease, liver disease ...) that indicate the condition. Because of the associative nature of the network, all lower nodes (cardio-vascular injury, endocarditis ...) are automatically included in the tree. The initial disease model is then used to set-up the candidate hypotheses and this directs the questioning of the user.

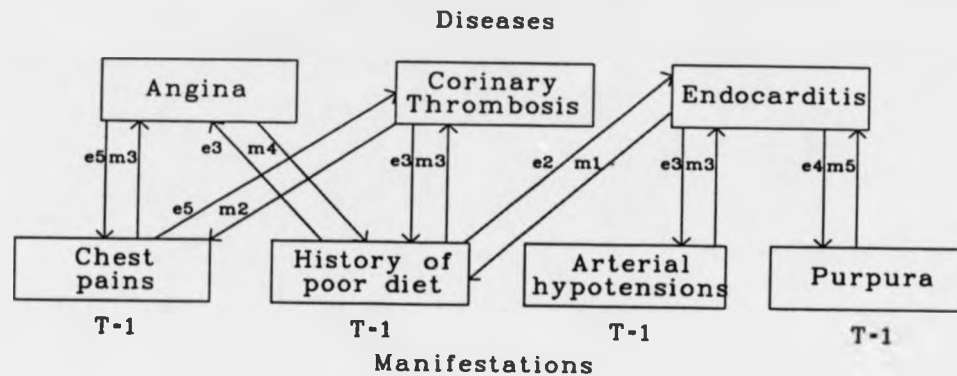


Figure 2.8 Sample Database Network for INTERNIST

The model has four lists of information compiled during a questioning session:

- a) A list of observed manifestations that do not relate to the disease network.
- b) A list of observed manifestations that relate to the disease network.
- c) A list of manifestations that have not been observed, but should be associated with the disease
- d) A list of manifestations associated with the disease model which should not have been observed.

Each disease node has the lists attached to it, and based on assessments the nodes are ranked in order of their likely association with the lists generated by the question

session. A RULEOUT or DISCRIMINATE strategy is then used to select the most promising hypotheses based on this ranking, and a final set of disease nodes selected for consideration. In general terms, the patient data is added to the system as a bottom-up process and then evaluated using additional manifestations that should be present given a specific prognosis, the later being a top-down process.

The main criticism of INTERNIST is that it operates in a serial manner, sometimes making the questioning slow and obvious. The clinician is often able to quickly change the line of questioning given key information, something that INTERNIST is slow to react to. This suggests that numerical ranking systems are not necessarily the most effective and responsive way of choosing between alternatives.

2.4.3 DENDRAL - a system for analysing chemical compounds (Buchanan and Feigenbaum 1978).

A problem can be defined in terms of a problem space with nodes representing states of a system and the links between them as actions that change the state of the system. The more complex the system the greater the number of alternative actions that can be applied to the present state and thus the greater the branching factor is said to be. It can also be said that the more complex the system the greater the number of changes in state (intermediate states) needed to transform the initial state (start state) into the required state (goal state). Figure 2.9 shows a problem tree for an unspecified problem illustrating the structure of this type of problem characterisation.

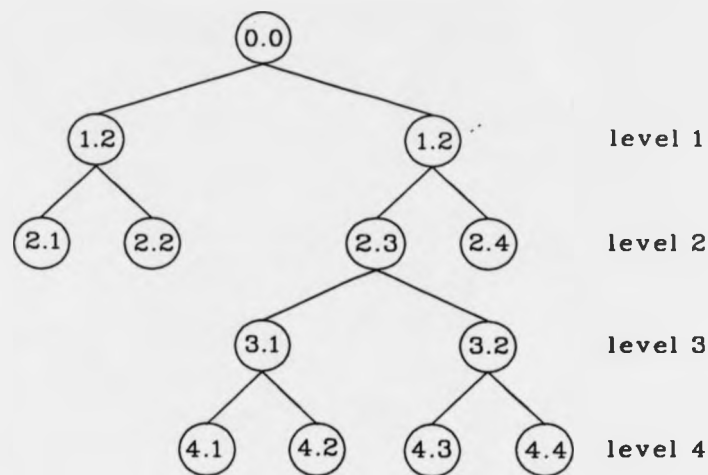


Figure 2.9 Problem Tree for Unspecified Problem

In Figure 2.9 the circles represent nodes and the links actions to achieve a change in state, and associated with each successive change is a level number or depth, and the breadth of the tree is indicated by the branching factor, which in this case is between 2-3. In numerical terms, the number of alternatives available is expressed as the branching factor (2-3) to the power of search (3-4). This means that to achieve a solution at level 4 it could require exploring $3^{**}4(81)$ alternative paths including backtracking. Faced with more complex problems, the choices soon become excessive and this is commonly referred to as the combinatorial explosion. The DENDRAL system is a complex domain and, therefore, has such problems, and has attempted to solve this by applying a weak heuristic method known as

generate-and-test to the analysis of X-ray crystallography data of unknown compounds. The method involves the generation of alternatives by expanding a single node, and then testing the new states by using a heuristic function to assess their plausibility.

In more detail, the knowledge of the system is stored in a rule base as a series of IF-THEN constructs, and by searching the knowledge base, DENDRAL can assess the evidence (data) from the analysis of the compound to decide whether the facts generated by the knowledge base are to be considered as valid or invalid. A generator (CONGEN) is used to generate a set of possible chemical structures, and through the application of constraints, limits the choice of possible structures from the test data. Three types of constraint are used.

- a) Graphical - symmetric structures are not unique.
- b) Syntactic - valencies will limit plausibility.
- c) Semantic - additional information of molecular tests.

After applying these constraints, a list of possible structures is generated and a list of impossible structures generated. The testing program now operates on the options produced by CONGEN using two programs, MSPRUNE and MSRANK. MSPRUNE takes each candidate chemical structure and generates a theoretical mass spectrum from it. This is compared to the test data and any structures that significantly deviate from the test data are pruned. The remaining candidate structures are ranked by MSRANK using

detailed knowledge of mass spectrometry to order the structures in accordance with predicted peaks in the test data, weighted in accordance with the importance of their presence. This information is then used to propose possible chemical structures for the test substance assuming each candidate passes the necessary threshold value of MSRANK. Figure 2.10 summarises the operation of DENDRAL.

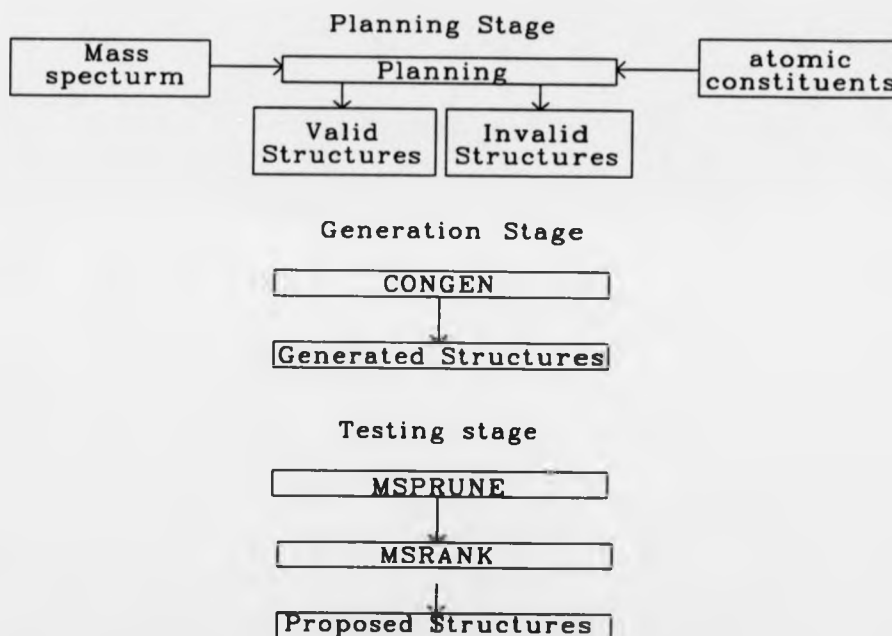


Figure 2.10 System Flowchart for DENDRAL

In criticising the DENDRAL project, it can be said that no attempt was made to model the expertise of the chemist. A static problem space was used with a generate-and-test

heuristic for limiting search. It also appears from transcripts that the system designers had great difficulty in encapsulating the information in a rule format, and that the knowledge elicitation phase of the project was especially complex. One possible solution to this problem is to build redundancy into the system by not attempting to account for details and employ a conflict resolution strategy to pick between competing rules. This would increase process time and reduce the accuracy of the system, but reduce the probability of error.

2.4.4 MOLGEN - a system to added in design of biological Experiments (Stefik 1980).

The final expert system considered here is MOLGEN, an expert system to assist biologists in the task of designing molecular experiments. As this type of system is concerned with design it cannot produce one correct solution, and the goals of the system cannot be specified from the start. This means that unlike with the other three expert systems, a simple heuristic function cannot be employed to reduce the search space of this problem domain. The only solution to the design of the system is to use methods of abstraction. The core of MOLGEN is represented as a triple layered structure with each level defined as a separate problem space with its own unique operators and problem characteristics. The three spaces are:

a) Strategy Space - this is concerned with meta-planning using general operators FOCUS, RESUME, GUESS, and UNDO, employing the basic search strategies required to manipulate the search space.

b) Design Space - this level is concerned with the experimental design and layout of the laboratory experiments, with both operators (REFINE, PROPOSE-GOAL, PROPAGATE-CONSTRAINTS) and objects (DIFFERENCE, CONSTRAINTS, REFINEMENT, TUPLE).

c) Laboratory Space - the space contains the necessary operators (SORT, MERGE, SCREEN, TRANSFORM) and objects (GENE, BACTERIUM, ENZYME, ANTIBIOTIC) necessary for gene splicing experiments.

When analysing a problem, these three spaces work together to form a hierarchical control structure. Entry to the control structure starts at the top level, and a planning strategy is selected. Two types of planning strategy are available:

a) Least Commitment

This has two operators available to it, FOCUS and RESUME and these are used to propose new planning stages and re-activate old ones

b) Heuristic Planning

This also has two operators, GUESS and UNDO. Heuristic planning is only used in situations where there is not sufficient information available to use Least Commitment.

The least commitment strategy has priority over the heuristic strategy as it provides the most effective planning procedures and is the initial starting point of the system. MOLGEN begins by communicating with the design space, requesting a task to FOCUS on. The tasks can be the proposal of a goal, the redefining of an operator or the propagation of constraints. The start point is usually the proposal of a goal, and once selected constraints can be applied to the problem space. If during this process a task cannot be FOCUSED on, often due to lack of constraints, then the current task is suspended and a new task found. Occasionally, a new task cannot be founded, and if all suspended tasks cannot be RESUMED, then the system returns to the top level to change the mode of operation from a least commitment strategy to a heuristic strategy. The GUESS operator is now used to select a plan in terms of an experimental design, this may be a standard experimental design based on the responses given by the user to the least commitment strategy. Once the top level plan has been formulated the second level strategy is again consulted and the steps in design considered. In the design space the three operators are used to act on the four objects in accordance with the responses of the user and the overall meta-planning strategy formulated. These produce a general

list of instructions for carrying out the experiments. For example, a goal may be proposed by the system (PROPOSE-GOAL) and this may be concerned with the REFINEMENT of an object in the laboratory space. Alternatively, the propagation of constraints (PROPAGATE-CONSTRAINT) may be required on a particular database item (TUPLE).

When the meta-planning objectives have been selected the system can fix on the details of the laboratory space to produce the exact stages in the experimental design. This is the planning stage of the design and outlines the steps required to carry out the molecular experiment. The production of a disease vaccine could be one objective, or the genetic engineering of a bacteria to produce protein for consumption could be another. The selection of the best design is not the objective of the system, more its to use the structure of the problem space to produce a good experimental design that will do what is required.

2.5 The Role of Expert Systems

Expert systems utilise the general problem solving characteristics of A.I. systems, and bind many of the algorithms of the latter into a formal architecture. This enables expert systems to have both an extensible and reusable software role across a range of problems. The inference engine is always domain independent and the accompanying representations are decompositional in nature (see chapter 4). If the expert system shell retains the capacity to decompose then a general problem solving

capacity results. There are commercial systems (NEXPERT GENSYM) currently available that can build representations for solving scheduling, diagnostic and classification problems. Tasks such as those solved by STRIPES and ABSTRIPES can be encoded into a general shell and solved using rules and formal logic.

Once domain knowledge is encoded into a system it can have a number of knowledge roles and these include:

- a) Solving problems without the intervention of an expert
- b) Teaching the naive user and sometimes the expert about the domain
- c) Organising knowledge and the benefits that arise from such record keeping (audit trails, case studies etc.)
- d) Extending expertise beyond the life-time of an expert

The capacity of the system to perform any of these roles is limited by the architecture. Generally speaking, the deeper the representation the greater the scope of the expert system.

2.6 Conclusions

Six A.I. systems have been outlined in this chapter, ranging from theoretical problem solving to abstract planning procedures. The trend has been to represent knowledge of the system in an increasingly structured way, and introduce a

diverse range of control strategies to drive them. Knowledge has been categorised in hierarchical schemes with the performance of the human expert serving as a valuable resource. Unfortunately, there emerges no clear design considerations indicating the best techniques for a specific problem. Most problem solving strategies are specific to the working domain, and not generalisable. Work on general problem solving has not proved successful, with the emphasis now being placed on the encodement and structuring of knowledge within an expert system architecture. This is the activity of modelling and may be the most promising general approach to problem solving.

Chapter 3

Control without Knowledge

3.1 Introduction

Expert systems are built to manipulate knowledge in relation to a current problem, or consultation. The architecture reflects this requirement (see Section 2.2). However, knowledge is not always available to direct processing towards the next stage in a consultation or 'solution procedure'. It is at these points that control without knowledge becomes important. The focus in this chapter will, therefore, be on the ways in which problems can be described using a "general" computational format, and how problems can be classified into different styles and solved using different control strategies. The use of the heuristic function will be analysed along with the processes of matching, constraint propagation, and search.

3.2 Problem Spaces

Problems can be understood in terms of a series of changing states connected together via actions that transform one state into another. The solving of a problem can then be seen in terms of selecting the appropriate actions to transform the initial state from one condition to another, creating a path through sets of alternatives to eventually achieve the solution or goal state. A number of alternative states that can be used to transform one state to another, and this is generally referred to as the breadth of the problem. The number of transformations of the states

required to achieve a solution is also important, and this is referred to as the depth of the problem. When trying to characterise the problem space for different domains the breadth and depth of the problem space will vary, a simple programme like naughts and crosses may require only a few alternatives and generate a very small search space, whilst trying to capture the essence of the chess domain may take many more alternatives. The graph in Figure 3.1 illustrates the levels of complexity for various problem domains, and gives an idea of the range and scale to problem solving.

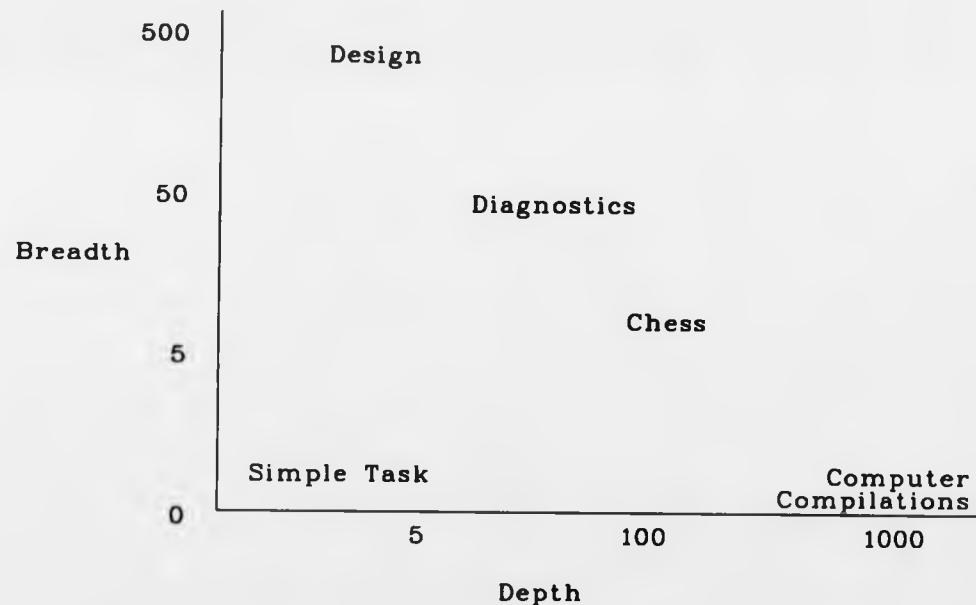


Figure 3.1 A Problem Space for Typical Problem Domains

The difficulty with problem solving is not so much one of identifying the scale of the search space, but the types of

search required. Different problems require different techniques to solve them, and there is not considered to be one general problem solving technique for all domains (Lenant 1982).

3.3 Classification of Problems

Problems can be divided into separate stages or intermediate states and the interdependency of the states determines how a problem might be classified. Some problems can be decomposed into simpler sub-problems and then a universal algorithm applied to solve them i.e. mathematical integration. However, some problems cannot be solved in this way and are considered non-decompositional. In other words, one change in state is dependant on another i.e. any manufacturing process.

A further method of defining a problem concerns the way the intermediate steps to solution from the initial state to the goal state or solution relate together. Three types of problem can be defined in this way:

- a) Ignorable - steps to solution can be ignored.
- b) Recoverable - steps to solution can be undone.
- c) Irrecoverable - steps to solution cannot be undone.

The control strategy necessary to organise search through the problem space is different for each of these problem types. Ignorable steps can be solved using simple recursive programming techniques that never backtrack. Recoverable steps can use a simple pushdown stack with items solved

(expanded and added to top of stack) in series from the top of the stack downwards. Irrecoverable steps require the most developed method of control, usually involving pre-planning to explore the steps through the problem space before carrying them out. The general principle is that the more recoverable the problem state the simpler the control strategy (Lenant 1982:20).

The certainty of the outcome of a problem is also an important parameter for classifying problems. The domain may not be predictable and it may not be possible to predict exactly what the results of a specific planning strategy will be. In these circumstances it is necessary to generate several plans or hypotheses and rate each of them as a means of choosing the best solution. In this respect, planning is like problem solving, but without feedback. It is an open-ended task and, therefore, needs some form of revision technique. For example, updating the plan with feedback from the environment to enable the rating of alternative plans.

Problem Type	Certain Outcome	Uncertain Outcome
Ignorable	Integration	Theorem Proving
Recoverable	Goal reasoning	Diagnostics
Irrecoverable	Compositional Problems	Game Playing

Table 3.1 Table of Problem Solving Characteristics.

Table 3.1 suggests a classification scheme for problem solving, combining the certainty of outcome with the recoverability of the process.

3.4 Search Strategy

Figure 3.1 illustrates how a problem can be described in terms of depth and breath. More specifically, a problem can also be described as a search tree, and Figure 3.2 illustrates a sample problem tree for an unspecified problem (Henson 1987).

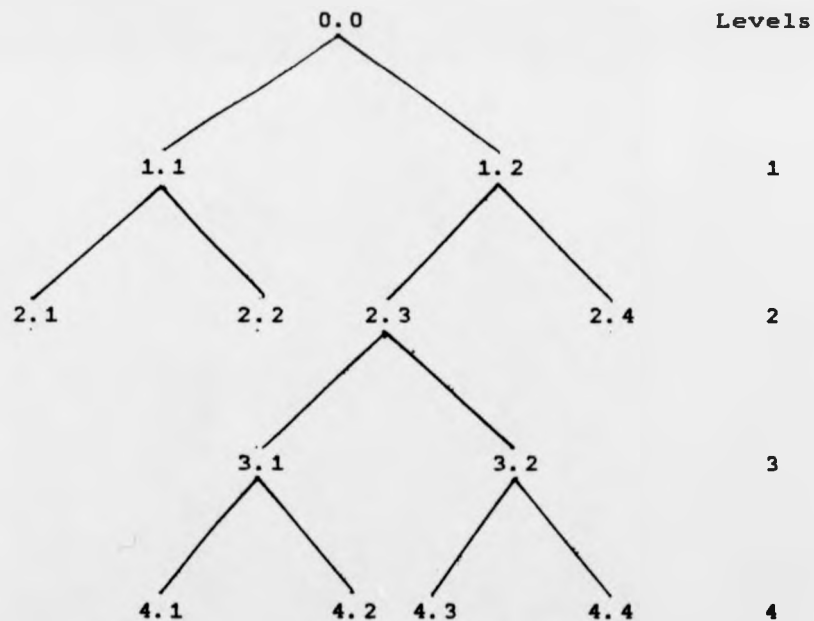


Figure 3.2 A Sample Problem Tree for Unspecified Problem

This method of characterisation implies that if all the alternatives to a problem domain could be specified, it would be possible to create a problem tree accounting for all the alternative states. It would then be possible to search the entire tree level by level to find the best solution for each domain specific problem. This is referred to as the brute force method of search. However, as the scale of the problems increases the number of alternatives expands at an exponential rate, meaning that the brute force method would take too long to find the best solution to a problem (Boden 1977). There are a number of other ways of applying search strategy, and each has advantages and disadvantages when applied to problem solving. These are:

- a) Breadth first search
- b) Depth first search
- c) Level first search
- d) Best first search

The breadth first search is conducted across the state space, systematically looking at all the alternatives at the top most level, comparing each to the goal state, before moving on to the next level of the tree to repeat the process until a match with the goal state is found. From Figure 3.2 this would mean a search path 1.1,1.2,2.1,2.2....4.2,4.3,4.4.

The depth search process is conducted down the problem tree, expanding nodes to their limits before returning back up the tree to expand the next alternative of the parent node.

This method of characterisation implies that if all the alternatives to a problem domain could be specified, it would be possible to create a problem tree accounting for all the alternative states. It would then be possible to search the entire tree level by level to find the best solution for each domain specific problem. This is referred to as the brute force method of search. However, as the scale of the problems increases the number of alternatives expands at an exponential rate, meaning that the brute force method would take too long to find the best solution to a problem (Boden 1977). There are a number of other ways of applying search strategy, and each has advantages and disadvantages when applied to problem solving. These are:

- a) Breadth first search
- b) Depth first search
- c) Level first search
- d) Best first search

The breadth first search is conducted across the state space, systematically looking at all the alternatives at the top most level, comparing each to the goal state, before moving on to the next level of the tree to repeat the process until a match with the goal state is found. From Figure 3.2 this would mean a search path 1.1,1.2,2.1,2.2....4.2,4.3,4.4.

The depth search process is conducted down the problem tree, expanding nodes to their limits before returning back up the tree to expand the next alternative of the parent node.

Comparing each state to the goal state, the search path for this strategy would be 1.1,2.1,2.2,...3.2,4.3,4.4.

The depth first search would tend to explore a small area of the problem space in detail, without looking at alternatives, whilst the breadth first search would consider all alternatives without exploring the problem in detail. Characteristically the depth first search is more likely to come to a solution too quickly and the breadth first search is likely to waste time considering irrelevant options before solving the problem. Both techniques are simple to implement, and the later is ideal for small problems.

As an alternative, the level strategy sets the depth of search to be considered, and only processes options lying within that problem space. The level could be set on the basis of the type of problem space, the search strategy being guided by the characteristics of the problem. Unlike the previous two search strategies, the level approach is applying a degree of top-down processing by modifying search behaviour as a function of the problem, giving the computer system an event horizon, beyond which the system knows nothing. If a level of 2 is set for this strategy the search process would be 1.1,2.1,2.2,1.2,2.3, and 2.4. If a solution is not found in this space then the level could be extended to include a larger area. This method is more complex to code, but more responsive to the problems within the domain. The best first search strategy, is in principle an extension of the level strategy, and expands nodes, applies an evaluation algorithm to the state to see how far it is from solution, and expands only the most promising paths.

Knowledge can now be used to constrain search and, hence, bring the domain factors into play. For example, in the chess domain, the propensity for a change in state to result in the loss of a valuable chess piece might constrain the search process (Berliner 1973). Equally, the possibility of controlling the centre of the board may also be used to limit alternatives and so on.

3.5 Direction of Search

The direction of search refers to the direction of reasoning within a problem domain and there are two ways of generating a solution path (Newell, Shaw, and Simon 1967):

- a) Backward Chaining - reasoning from goal to start state
- b) Forward Chaining - reasoning from start to goal state

Backward chaining refers to a process of building a sequence of events that might be a solution to the problem by starting with the goal configuration(s) at the root on tree at level (n), generating the next level (n+1) of the tree by finding all the states who have consequences of actions that match the root node level (n), and then using the conditions of those actions to generate the next level of the tree (n+2) by matching their conditions at level (n+1) to all the consequences of action that match at the next state. This process is repeated until the initial conditions are matched to the generated conditions.

Forward chaining operates in the opposite direction, and is referred to as data driven reasoning since it uses the

information at the start of the problem to constrain the expansion of nodes that match the start state. The process begins by building a sequence of events that might be a solution to the problem, starting with the initial configuration at the root of the tree at level (n), generating the next level of the tree (n+1) by finding all the states whose conditions match the root node, and then using the consequences of action to generate the next level of nodes. The states whose conditions match the root node (n+1) are again used to create new consequences of actions, and the cycle is repeated until the goal conditions match. Both direction of search can be used in problem solving, but the best strategy depends on the characteristics of the problem space. There are three considerations for deciding on the reasoning direction:

- a) Ratio of start states to goal states.
- b) The tendency of the branching factor.
- c) The needs for explaining reasoning.

It is generally considered easier to move from a small set of start states to a large set of goal states, or move from a small set of goal states to a larger set of start states. In both cases it is always better to move towards the bigger target. In this respect, backward chaining is best used when the goal set is smallest since the reasoning is moving towards a larger target. However, if there are a small number of start states and a very large number of goal states then forward chaining is preferred.

The branching factor also has an influence on the direction of search. The general rule here is that it is always better to move in the direction with a lowering branching factor. This means that if the branching factor is largest going from start \rightarrow goal then use backward chaining, and if the branching factor is smallest going from start \rightarrow goal then use forward chaining.

Finally, if it is necessary to justify the reasoning process of the system, then it is often better to work from the goal backwards, hence the preference for using backward chaining. As a compromise, a bilateral search method can be used, sometimes referred to as sideways chaining or bi-directional search (Hewitt 1971), and here a mixture of forward chaining and backward chaining is used. With an uninformed search strategy i.e. without the guidance of knowledge, sideways chaining is useful. However, there is a possibility of this method of search failing.

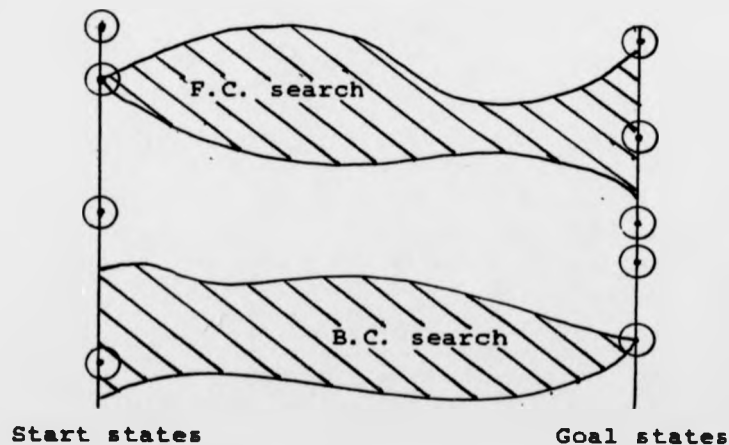


Figure 3.3 The Problem of the Sideways Chaining Method

Figure 3.3 shows how double the effort can be expended if the two paths fail to meet during the reasoning process. Under these circumstances forward or backward chaining would be less expensive (Pohl 1971). The use of sideways chaining can help when the problem illustrated in Figure 3.3 is overcome, and the PLANNER language showed that by monitoring stages in both directions the 'miss effect' could be controlled (Hewitt 1971).

However, it is suggested that the more informed the search, the less the value of complex reasoning strategies.

3.5 Heuristic Search

The search processes outlined above are general bottom-up processes taking no account of either the current state of the system as compared to the overall goals, or possible knowledge about the problem domain that could be used in specifying context. Heuristic search strategy uses a top-down approach to problem solving suggesting the use of common sense or general heuristics to limit the search space. There are two widely used methods of applying heuristics:

a) Incorporate special purpose rules into the domain ie in the chess domain define not just the legal moves, but the sensible moves (Berliner 1973). This might be referred to as the development of a knowledge base.

b) Apply a function that evaluates individual problem states and determines how appropriate they are (Newell, Shaw, and

Simon 1967). This process takes account of the dynamics of the problem and utilises context to limit search.

Heuristic functions map on to the problem state a measure of acceptability, possibly in the form of a numerical analysis and the heuristic uses rules to maximise or minimise this acceptability as a means of guiding the search behaviour of a system. In chess, this might mean the attachment of a simple 'material advantage value' to each of the possible nodes in a problem tree, selecting those paths with the best score. To enhance search behaviour in this way requires the specification of three important factors:

- a) Representation
- b) Matching procedures
- c) Conflict Resolution

3.5.1 Representation

Problems can be characterised as problem trees or problem graphs, with each node represented as a point in the problem space. The use of heuristics requires that the node is represented in a way that allows evaluating actions to be carried out on it. The problem is that if all the information of the domain were to be stored at every node then the descriptions of the current state would be unacceptably long. A way of labelling those items that change is, therefore, required, but in a way that preserves the values at other nodes. This is known as the frame problem (McCarthy and Hayes 1969). If only changes are

recorded at each node, then the system operates effectively until the search strategy has to backtrack and seek an alternative path, the exception being monotonic reasoning. Three solutions to this problem are possible. Firstly, do not modify the initial state description, but instead store at each node the changes to be made. This is a type of planning procedure used in many A.I. systems such as NOAH (Sacerdoti 1975). Secondly, modify the initial state description, but store at each node instructions on what to do if backtracking is required. This is like storing plans at each node to restore the initial state description if the current path does not appear to lead to a solution (Sacerdoti 1974). Finally, use a state variable to indicate when the facts are true and use this like a date stamp (Doyle 1979).

3.5.2 Matching Procedure

When knowledge is represented in a system, it is probable that at various times during the search process elements of the knowledge base will be required to guide the reasoning. Matching refers to the process of selecting those elements in the knowledge base that can be used to guide search. In other words, it can be defined as the method used to extract from a closed collection of rules, those that apply to a given point in the search space (Forgy 1983). Matching is especially important because often many elements will satisfy the initial requirements of the current state, and a selection process is required to reduce the burden on the heuristics (see Section 7.3.1).

3.5.3 Conflict Resolution

In applying knowledge to a solution, many rules may exist in the knowledge base and can be applied to the current problem state. Conflict resolution is the method used to decide in which order the rules should be matched to the current state (Newell 1973). As a guide to which rules to apply first, when matching against a key pattern, it is better to select rules with keys that occur with less frequency in the knowledge base than those that are more common. It is also better to select rules for matching that have most recently been used rather than those further down the stack. This is analogous to the modelling of behaviour in human short term memory (see Section 7.1.2).

3.6 Types of Heuristic Search (Weak Methods).

In applying heuristics to the search process, the problem domain must be configured in such a way that knowledge is represented in a structured fashion, and that procedures exist for accessing that knowledge using a priority system with matching functions to select knowledge segments applicable to the current state of the problem. It is also important that a control strategy is used when applying such heuristics and three methods are outlined:

- a) Generate-and-Test (Lindsay, Buchana, and Feigenbaum 1980)
- b) Hill Climbing (Lenat 1982)
- c) Best First (Martelli, and Montanari 1978)

Each of these three procedures can be used to direct the processing of the system, and are used to control the goal directing mechanisms of the search strategy. In principle, the heuristic applies an algorithm to the current state in the problem space, and this is used to evaluate the position in terms of distance from a goal, distance from the start state, and comparative values generated from alternative positions in the search space. Then, based on the results of this analysis, the heuristic can be used to decide which of the available routes is the best one to expand.

3.6.1 Generate-and-Test

This is a very simple heuristic operating on a depth first principle. Backtracking can be employed as method of retrieving previous states in the system, and this improves the performance of the heuristics. The generate-and-test method works by generating possible solutions in the problem space from the current state. These viable options are then tested using knowledge from the domain to restrict the problem states. Those problem states that are left are matched to see if a solution can be found, and if not repeated using a different branch of the problem space until a match is found. Figure 3.4 presents a block flow diagram of the method.

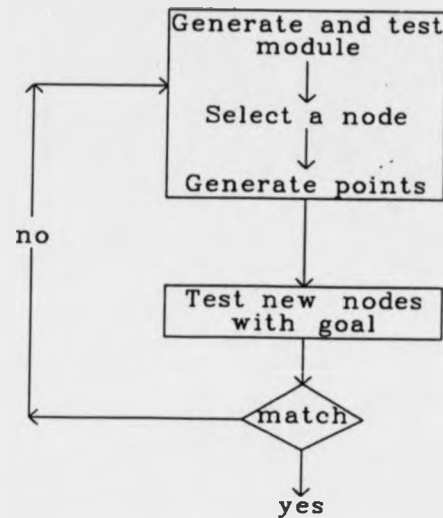


Figure 3.4 Flow Diagram for Generate-and-Test Heuristic

With this system, the event horizon can be pushed well back, but due to its very nature the resultant search space for this type of control strategy is very small. Such a system is more likely to miss the solution than a broader search method. However, the DENDRAL project successfully used a modified version of this heuristic called plan-generate-test within an expert system architecture (Fikes, Hart, and Nilsson 1972).

3.6.2 Hill Climbing

This heuristic is an extension of the generate-and-test method and employs a process of continuously comparing the current state with the goal state in order to help determine the path through the search space. The basic G-T system is used at the start of the process, but when a match is not

possible the system selects all the rules applicable to the solution, and tests each element for its distance from the goal, and if a match is still not found the best solution is used to generate the next level of nodes. This process continues until a match is found with the goal. Figure 3.5 summarise the system.

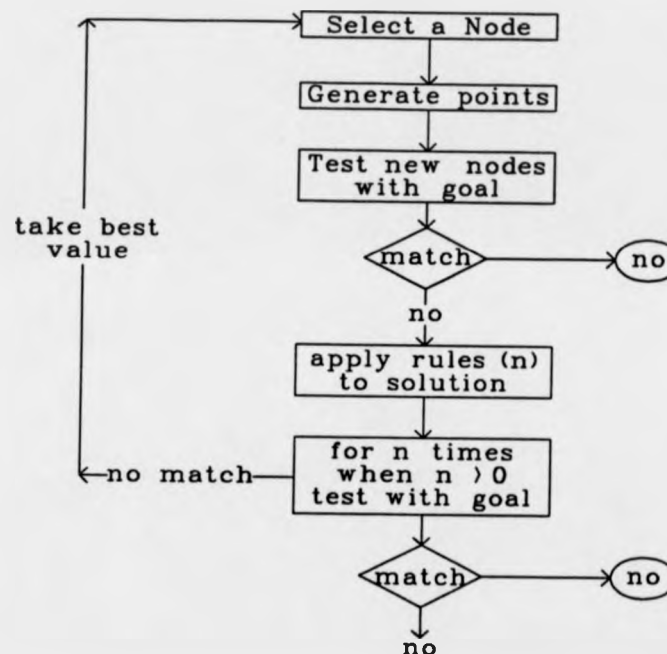


Figure 3.5 Flow Diagram for Hill Climbing Heuristic

This approach has advantages over the G-T system since it uses feedback to guide the search. However, by assigning values to different points in the problem space, and then

comparing these to the highest value in the solution, problems such as the local maximum, the plateau and ridge can be avoided. (Boden 1977).

Although these specific problems can be overcome by modifying the control strategy, the hill climbing heuristic is still essentially a depth first approach with the same difficulties as the G-T method. If the problem space is very uneven then Hillclimbing may fail to find the solution.

3.6.3 Best First

Unlike the previous methods, this is a mixed method of search, using both depth first and breadth first strategies. It involves expanding the most promising node of the search space first, and continuing to pursue this path whilst the values at the selected node are better than the unexpanded node(s). If any unexpanded node subsequently appears more promising than the current search path, it is memorised and then left, with the new path being generated until either a solution is found, or the original path again appears better, or a second unexpanded node becomes the most favoured. The A* algorithm is used in this method to calculate the values for the nodes, and is a graphical control strategy that classifies nodes OPEN for those yet to be explored and CLOSED for those already expanded (A*). All closed nodes have computed values that correspond to the BEST NODE selection process and the values assigned to the node(s) is the combination of the known cost of getting from the start state to the current state and the estimated cost

of getting from the current state to the goal state. The A* algorithm assumes an independence between paths to solution and computes each route as an alternative. Figure 3.6 illustrates the Best Path heuristic.

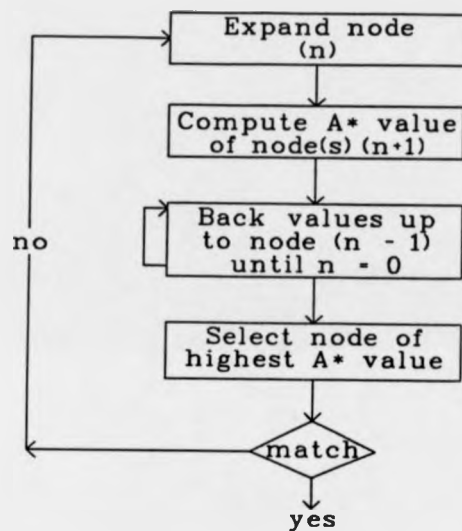


Figure 3.6 Flow diagram for Best Path Heuristic

The system illustrated here is for paths that are assumed to be independent, although this is not always the case. A change in the current state can often change the states of other areas in the problem domain previously calculated, and therefore to cope with interdependencies other solutions are required.

3.7 The Use of Constraints

Constraints refer to the application of domain specific knowledge to the problem domain, limiting the role of search in the process of problem solving (Fikes 1970). Search is only really necessary when the problem domain is unstructured, and should only be used as a back-up procedure when knowledgeable solutions fail. Constraints can be used to structure the search space and can define the necessary conditions that need to be met to satisfy the goal state. In other words, constraints could be seen as a way of dealing with multiple goals, and, are therefore, closer to representing real world problems such as those solved by expert system technology.

By structuring constraints as a list of necessary conditions, and then manipulating that list as various elements of the problem are solved, it becomes possible to modify the search process directly by applying inferencing rules that generate contradictions indicating that either a partial solution has been found requiring further exploration, or that the goal state has been reached. The use of constraints within the problem space can be seen as a 'set of problems' with a 'set of possible solutions'. By defining multiple conditions in the form of a list, the aim of the search process is redefined as the shortening of the list by satisfying the conditions placed on the search process. A decreasing set of constraints is an indication that the search process is proceeding in the right direction, whereas an expanding list implies the opposite.

3.8 Conclusions

Problem characteristics are of direct relevance to expert system design. They suggest ways in which the search space of a problem domain may be limited given the assumption that it is not always possible to make knowledgeable choices about the problem at hand. These methods are especially useful when data are missing from the process, or values are unknown. The techniques used are often described as being "weak", but they are usually necessary because it is improbable that powerful enough logics exist to maintain a consultation throughout. The next chapter examines the control of a consultation through knowledge, showing the aspects of representation and inference that are central to the expert system structure.

Chapter 4

Control with Knowledge

4.1 Introduction

The architecture of an expert system is designed to utilise knowledge represented in the system through the use of an inferencing mechanism or engine. A knowledge base holds expertise in a structured form that is independent of the procedures used to access it. The inference engine uses the stored knowledge to infer new knowledge required to solve problems (see Section 2.2 for overview). These two components are the core of an expert system, and form the process of control with knowledge. The knowledge in the system can be represented in many different ways, and utilised using different types of inference. Since the core has structural independence, it is possible to mix representations and methods of inference. The type of mix used depends on the characteristics of the expert system domain. More than one representation and inference engine can be used, and in fact many successful expert systems use complex combinations in their core (see Section 2.4 for expert systems).

4.2 Knowledge Representation

Broadly speaking, there are three approaches to knowledge representation: the logicians approach; the object orientated approach; and the action orientated approach. The first utilises logical systems of control such as predicate and propositional logic, and gives rise to the

use of logic programming, and in particular the symbolic manipulation of objects using PROLOG (Kowalski 1979). In contrast, the object orientated approach is based on the principles of frames and semantic networks (Minsky 1974, Woods 1975). These methods centre on the collection of facts around a defined object connected together into a networked structure using binary relations. In a programming sense, frames and semantic nets are similar to each other, but differ in the way they connect conceptual objects together. Object sensitive programs can be written to act on the object structure, making system development both reusable and modular. The action orientated approach includes production rule methodology and relational databases (Ullman 1982). The aim of these representations are to fire rules when conditions are met, hence the term action. Production rules are usually in an IF ... THEN ... format. Relational databases can use program references stored in field locations to fire procedures.

4.2.1 Logical Representations

Logical representation is a context free expression of knowledge. It is domain independent, with a well defined syntax. The symbols of this logic are variables, predicate constants, and connectives. Objects can be either variables or constants and are combined with relations to form propositions such as:

ISA(curve, smooth)

In this example, the proposition is composed of an ISA relation with two arguments or objects, curve and smooth. The individual propositions are referred to as atomic propositions and can be combined together using the logical connectives to form complex expressions (see Table 4.1).

Name	Symbol	Description
Negation	\neg	NOT
Conjunction	\wedge	AND
Disconjunction	\vee	OR
Implication	\rightarrow	THEN
Equivalence	$=$	EQUALS

Table 4.1 The Logical Connectives for logic systems

When the atomic propositions are combined together they are referred to as logical propositions, and in theory, almost all statements can be expressed using this logical representation. For example, in the field of X-ray rocking curve analysis, the complex expression "If the simulated rocking curve (SRC) mismatches the experimental rocking curve (ERC) then resimulated unless the S/N ratio is above the threshold, in which case apply the clean-up algorithm" can be expressed as:

$\sim \text{Match}(\text{SRC}, \text{ERC}) \wedge \sim \text{Above}(\text{S}/\text{N}, \text{Threshold}) \rightarrow \text{Call}(\text{Simulate})$
 $\wedge \text{Above}(\text{S}/\text{N}, \text{Threshold}) \rightarrow \text{Clean-up}(\text{ERC}) .$

In this example, Match, Above, Call and Clean-up are all predicates, and SRC, ERC, S/N, Threshold and Simulate are all arguments or objects. They are formed into two propositions [Match(SRC,ERC), Above(S/N,Threshold)] implying the proposition [Call(Simulate)], and the [Above(S/N, Threshold)] proposition implying the [Clean-up(ERC)] proposition.

The overall success of logic in representing the world is dependant on the decisions made during the formulation of the domain. Logic is an object language, whereas English is a meta-language, and representations of the later relies on the translation of informal statements into formal object language. In representing the domain it is, therefore, necessary to consider the following points:

- a) The number of actions or predicates required
- b) The number of arguments per predicate
- c) How to build functions in respect of size and number
- d) The constants of the system

Given these requirements it is possible to build expert systems using such a representation. However, it is important to regard the construction of such domain rules specified in this report as different from the inferencing rules that drive the system. The domain rules constitute the knowledge base of the expert system whereas the inferencing

rules constitute the inference engine of the expert system. The rules of inference are the instructions on how to use knowledge stored in the expert system (see Section 4.3). The binary pair forms the basis of representing knowledge defined in terms of logic. By structuring knowledge in a formal way very simple control mechanisms or rules of inference can be applied to the system as a proof procedure. Unfortunately, most proof procedures have problems dealing with 'common sense' reasoning since once a conclusion has been reached it is impossible to withdraw it, even when additional information is discovered. In this respect, most logical systems are monotonic. There are two ways of dealing with this major problem. One is to generate general problem solving proofs, but as stated in the introduction these have not proved successful (Newell and Simon 1963). The other is the use of high level programming languages such as PROLOG which provides one method of proof called resolution (see Section 4.3). This later method seems to be gaining popularity as a method for building expert systems. However, there are further alternatives to the logic methods, and researchers have looked at the development of forms of representation most of which are based on 'cognitively compatible' structures.

4.2.2 Semantic Networks

The notation of the semantic network is based on the idea of an associative memory, and was first suggested as a form of knowledge representation by Quillian (1968). The main

characteristic of the semantic network is that links or pointers are used to connect individual facts together into a networked structure. This structure is the root of knowledge representation in the system. The functional unit of the structure is a binodal connection consisting of two nodes joined together by a link. The nodes represent names of objects that can have attributes associated with them, and the links represent a directional relationship between the objects. Figure 4.1 gives an example of a functional unit which could also be seen as a binary predicate, the nodes being equivalent to the arguments, and the predicate representing the link.



Figure 4.1 The Basic Binodal Unit of a Semantic Network

Figure 4.1 expresses the fact that a simulated curve matches an experimental curve. The direction of the link indicates that the simulated curve is matched to the experimental curve, thus sustaining the subject and object relationship between each fact. This is especially important in respect of the linguistic representation of information (Fillmore 1968).

A network is built-up of groups of these binodal units, with each node linking to as many other nodes as necessary. The connections between nodes can be arranged to form a highly structured network and the ISA or TYPEOF or MADEOF relations suggest membership of one object to another either in the form of object composition as in TYPEOF and MADEOF, or in the form of an object hierarchy as in the ISA relation. As each node can have attributes associated with it, it's possible to associate particular types of knowledge with specific nodes in the hierarchy, and selectively combine information according to the relations between the nodes. In the case of a network hierarchy, the ISA relation is a property inheriting link between nodes and allows information associated with a higher order node to be automatically transferred to its successors. This saves on the space required to store knowledge in the network, but demands a rigorous knowledge structure. Figure 4.2 gives an example of the use of the relations in the domain of X-ray crystallography, showing the theoretical representation of the node for deformation.

Figure 4.2 illustrates the node Deformation, which is associated with a Rocking Curve and a Sample. The deformation is the result of combining these two properties and is composed of a Crystal Lattice which automatically inherits the same experimental paradigms as the former node. Experimental Results is a high order node which inherits the unique properties of deformation, and in this example would probably have many other nodes associated with it.

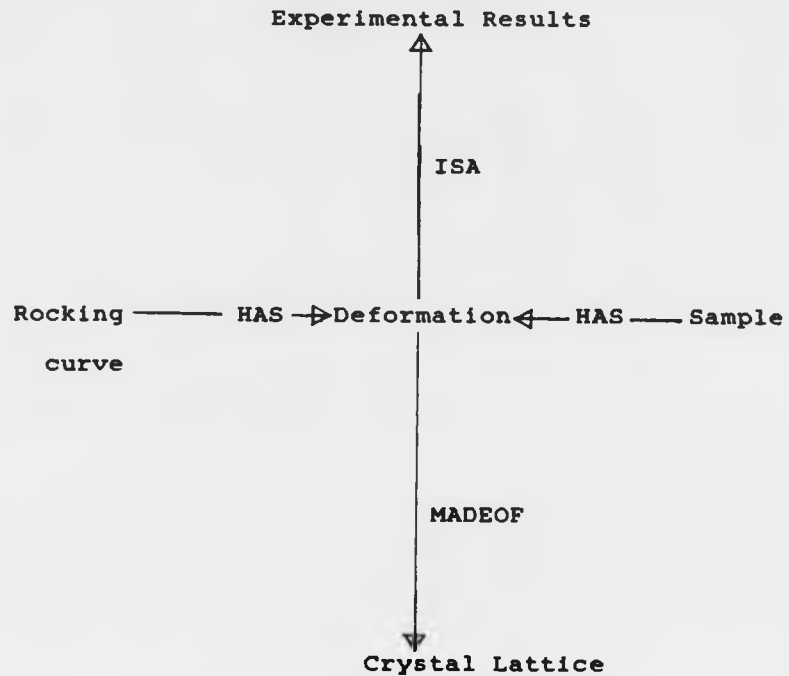


Figure 4.2 Semantic Network for the Node Deformation.

The INTERNIST project developed at the University of Pittsburgh is an example of the use of property inheriting networks based on these ideas, modelled within a expert system architecture (Miller, Pople, and Myers 1982).

In some respects, semantic networks are similar to predicate logic, they have a bipolar structure that resembles the binary predicate, and each functional unit of the network can be linked via further relations to build a network which are the same as the logical connectives used to join propositions into compound expressions. However, semantic

networks suggest that knowledge of objects is gathered in a specific area, which means that having found an object it is possible to access information on it. In computing terms, such an object orientated approach would use indexing to help in the organisation of the knowledge base in the same way as indexing helps in the organisation of a database.

4.2.3 Frames

Minsky was the first cognitive researcher to suggest the idea of a frame structure for knowledge representation (Minsky 1974). The frame is a structure in memory that is used to fit the problem to the context by altering the way in which the situation is seen. The frame is a data structure that represents this situation in a prototypical sense, and has three types of information attached to it:

- a) Information about the utilisation of the frame.
- b) Information about what to do given certain expectations of the situation.
- c) Information about what to do if these expectations are unfulfilled.

In structural terms, a frame can be seen as a type of semantic network, with node and interconnecting relations. The nodes are hierarchically arranged with those at the top fixed and always true, and those at the bottom possibly variable, being referred to as slots filled by specific

instances of data. The terminal levels of the network structure may specify conditions to be met for the frame to be instantiated, and these conditions will be either smaller sub-frames or objects with a specific value. Each frame is linked to others as a frame system which can be used to specify viewpoints of a visual scene, the cause-and-effect relations of an action, or the changes in a conceptual viewpoint.

Figure 4.3 gives an example of how part of the X-ray crystallography domain might be represented, showing the frame structure for the general concept 'experimental curve

```
name:          EXPERIMENTAL CURVE
               Type-of:   TEST RESULT
               code: .....
               lattice-type: .....
               simulation-curve: ....
               date: .....
```

Figure 4.3 A Frame Outline of Concept Experimental Curve.

In Figure 4.3 the frame has a name that refers to the concept it identifies, in this case EXPERIMENTAL CURVE. The rest of the frame is made up of descriptions that are attributes of the frame and these are termed slots. The slots are the basic structural elements of the concept, and have slot values which are filled when an object is matched

to the concept. This process is known as instantiation. Only if a complete match is made is the frame instantiated. In Figure 4.3 the first slot 'type-of' references a higher order concept called TEST RESULTS, and information about this frame refers to a sub-set of information called EXPERIMENTAL CURVES. There may be other test results relevant to the domain, and EXPERIMENTAL CURVES will be one type of many. Activating the TEST RESULTS may require more general information about other test results that may either confirm or contradict information extracted by the EXPERIMENTAL CURVE frame. Code is the next slot and this is a unit value that could be used as a key to identify a specific instance of the concept. The next slot, lattice-type, would specify the general type of deformation of the crystal structure with more detailed information stored in the sub-frame LATTICE. Simulation-curve could be a further sub-frame, linking procedures that will eventually match the simulated curve to the experimental curve. Finally, a date stamp could be used in the last slot as a cataloging method. Any number of slots can be used for each frame and the number used will be determined by the number of attributes and links required to specify the concept. Figure 4.4 gives an example of a specific object concept from the general frame structure (see Figure 4.3), and shows what attributes might be specified.

In Figure 4.4, TEST RESULT, LATTICE and SIMULATION-CURVE refer to other frames in the frame structure of the domain.

name: EXPERIMENTAL CURVE
Type-of: TEST RESULT
code: (contract no. XXX, serial no. XXXXX)
lattice-type: LATTICE
simulation-curve: SIMULATION-CURVE
date: (month, year) (default: now)

Figure 4.4 Object Concept for Frame Experimental Curves.

The frames are linked hierarchically so that some frames are super-ordinate like TEST RESULTS, some are para-ordinate like SIMULATION CURVES, and others are sub-ordinate like LATTICE. The sub-ordinate frames will inherit the characteristics of the calling frame, whilst the calling frame will restrict the classification or ranges of information to the sub-frame.

The units, such as the date and code slots, offer restrictions on what slot fillers qualify for inclusion, and these may be specified as constants or ranges between certain values. Figure 4.4 shows that code and date are both units, and could be used to drive a question program for input to the domain database. The date has a default value of 'now' attached to the slot, and this is assigned in the absence of information.

If all the information in the frame is within the constraints of the system then the slots are filled with a specific instance of an object, and the frame instantiated.

The results of the frame are then either added to a database or are used by other frames in the system until such point as the original inquiry is satisfied. In Figure 4.5 an instantiated frame is shown with all the slots filled with values. This is the final process of the frame system.

```
name:          exp-1
               Type-of:  TEST RESULT
               code: (contract no. C45, serial no. 40003)
               lattice-type: lat-1
               simulation-curve: sim-1
               date: (may-1988)
```

Figure 4.5 An Instantiated Frame for Experimental Curve.

It will be noticed from Figure 4.5 that the name of the frame is now unique, with an identifiable suffix that is also translated to those slots whose values lie outside the frame, thus the lattice type and simulation curve both have a suffix of 1 to identify this information in association with this frame.

The frame is a useful general purpose structure for representing knowledge. It is very much like the semantic network both in terms of its structure and the way in which it is programmed into a system. The frame has advantages over other forms of representation when it comes to viewing

a situation from an unrecognised position, and this has proved especially helpful in the domain of visual information processing [63]. The frame can also be partially completed and in this case either defaults can be applied, or if not available, matches made upwards through the frame hierarchy until a general enough frame is found to fit the problem.

4.2.4 Production Rules

Knowledge represented as a Production rule is the most commonly used format in the expert system environment. The aim of a production rule system is to write procedures that form a sequence of rules to solve a problem. It is a very simple way of representing knowledge, and closely aligned to the more conventional forms of procedural programming. The basic unit of the production rule system is the IF ... THEN ... rule and this takes the general form:

IF condition THEN action.

A rule is formed from this structure and knowledge can be represented as a series of such units bound together by a control structure. As many conditions or actions can be assigned to each rule the extent of change incurred by invoking the rule can be governed by its size. The rule works by producing an action, which could set anything from a global variable to a switch, but on condition that certain factors are satisfied. Technically, the conditions for

invoking the rule on the left hand side is called the antecedent and the actions resulting from its application on the right hand side is called the consequent, and if the antecedent is satisfied then the rule fires and the consequent follows, but if the antecedent is not satisfied then the rule remains static. Overall, there are three elements in a production rule system:

- a) Knowledge base - representing the rules of the domain.
- b) Database - representing the current problem state.
- c) Control Structure - deciding which rule to apply next.

4.2.5 Knowledge Representation Overview

In this section four types of knowledge representation have been summarised, using examples from the X-ray crystallography domain. Each method has certain applications, and it can be generally said that predicate logic lends itself best to more formal type of representation especially domains that require only monotonic reasoning. The object orientated approach of frames and semantic networks is better suited to representing knowledge that has a cognitive style (classification or groupings), and indeed, both methods develop strength from their foundations in cognitive research. Frames has proved useful for representing the visual world, and the principles of semantic networks has been used in expert systems (Kulikowski and Weiss 1984). The production rule system is the most favoured representation technique for expert systems, perhaps due to its associated

links with procedural programming. It is also generally accepted that rules are better than other type of representation when facts are poorly structured and isolated (Buchanan and Shortliffe 1984).

All expert systems use one and sometimes combinations of these four main representation schemes, but certain methods are better than other at handling specific tasks. It is, therefore, important to selected the most appropriate method or combination of methods for representation during the design phase of expert system development.

4.3 Methods of Inference

At the most general level, knowledge is itself a passive medium, and has no influence on events. It is only the application of knowledge to the solving of problems that results in its dynamic qualities. This process could be compared to the application of data to the task of providing input to the programming of a solution, thereby, resulting in its transformation into information through contextual use. In a similar manner, methods of inference are the ways in which knowledge is employed to solve a problem, with the framework of an expert system providing the contextual restraints of domain dependency. More specifically, inference is the process of producing new facts or rules from a given set of facts and rules through inference. A given set of facts and rules would be the initial starting conditioning of a problem. Typically, these would be stored in a database, a goal would be set, and methods of inference

would be applied to this starting condition using the knowledge of a restricted domain as a 'matching mechanism' to infer new rules and facts for storage in the database. These new rules and facts could then be used, along with the starting conditions, to infer yet more new rules and facts about the problem, and the process cyclically continued until the conditions in the database match the goal state. To achieve these results, an inference engine can use many methods to drive the knowledge base of an expert system. The type of method used and degree of inference will depend on the way in which the knowledge of the system has been encoded. For example, with predicate logic there are two distinct types of rule, the domain rules (the binary predicates) which are the elements of knowledge, and the inference rules, and through the application of the later rules to the former rules new facts can be evolved. This distinction is retained with the production rule system. The knowledge base and control structure are both clearly defined in a production rule system with the domain rules kept in isolation from each other, with the control structure interacting with the rules. However, with object orientated systems such as frames and semantic networks the connections between the elements, ie the frames and binodal links respectively, cannot be processed in isolation. Both representations joined together there elements into a highly structured framework, and much of the expertise in the system is present in the structure of the knowledge base. In this case inference plays less of a role.

There are different types of inference and the two most commonly used principles are:

- a) Logical Inference
- b) Statistical Inference

Logical inference is the application of the general statements about the relationships between assumptions and conclusions. Statistical Inference is the principle of evaluating the evidence against the hypothesis in the selection of the strongest causal link. Both of these methods attempt to deduce what the solution to a problem might be, but use fundamentally different approaches to the driving of the knowledge base.

4.3.1 Logical Inference

This principle is based on a set of well defined rules of inference, that when applied to facts in a database yield new information that is always valid. The rules of inference must be applied to correctly formed propositions or well-formed formulas (wffs) which is governed by the rules of formation (see Section 4.2). For example, the connective ~ (not) must always be placed in front of the target proposition ie ~match(sim_1,expt_2) which means no match between sim_1 and expt_2, or the connective can only be placed between two propositions ie match(sim_1,expt_3) ~match(sim_2,expt_3) which means that there is only one match for expt_3 with sim_1. In both examples sim_x could refer to an instance of a simulated curve and expt_x and an

instance of an experimental curve from the X-ray crystallography domain. Table 4.2 gives a summary of six rules of inference most commonly used.

Rule	Result	Example
Modus ponendo ponens (MPP).	$A \rightarrow B, A \vdash B$	IF The crystal is etched THEN test, The crystal is etched THEREFORE test.
Modus tollendo tollens (MTT).	$A \rightarrow B, \neg B \vdash \neg A$	IF The crystal is etched THEN test, The crystal NOT etched THEREFORE do NOT test.
Double Negation (DN)	$A \vdash \neg(\neg A)$	The crystal is etched THEREFORE The crystal is NOT NOT etched.
Andintroduction (AINT).	$A, B \vdash (A \& B)$	The crystal is etched, it has been tested THEREFORE The crystal is etched AND tested.
Reductio ad absurdum (RAA).	$A \rightarrow B, A \rightarrow \neg B \vdash \neg A$	IF The crystal is etched THEN test, IF The crystal is etched THEN do NOT test THEREFORE The crystal is not etched.
Universal specialism (US).	$(\forall X) W(X), A \vdash W(A)$	All objects that are crystals have deformations, the sample is crystal THEREFORE the sample has deformations.

Table 4.2 The Basic Rules of Logical Inference.

The rules of inference can be applied to facts in the database to deduce whether certain conditions are true or otherwise. To illustrate the operation of these rules consider the following conditions. Firstly, that there is a general understanding which says that when a crystal has been etched, no large deformations will be found in the sample, which could be considered a rule in the knowledge base. Secondly, a fact asserted in the database which states that large deformations have been found in the sample. Thirdly, a general query is set-up to prove that the crystal has not been etched. Converting each of these three items into predicate logic we get:

Rule: `etched(crystal) -> ~deformations(crystal,sample)`

Fact: `deformations(crystal,sample)`

Query: `~etched(crystal).`

Using the rules of inference the query becomes the goal state of the system, the fact becomes the start state of the system, the rule becomes the domain dependant pattern matcher, and the rules of inference (Table 4.2) the control structure. The task is to apply the inferencing rules to either the start state or the goal state, transforming either to match the left or right side of the domain rule thereby proving the query.

In this example the rule MPP yields no match when applied to either the fact or query. However, the DN rule can be applied to the fact to give an intermediate result:

Fact: deformations(crystal,sample)

The DN rule can prove that deformations(crystal,sample) is equivalent to $\neg(\neg\text{deformations}(\text{crystal},\text{sample}))$ therefore applying DN rule yields the equivalent fact:

Fact: $\neg(\neg\text{deformations}(\text{crystal},\text{sample}))$

The bracketed part of the fact now matches the RHS of the domain rule ie

Rule: etched(crystal) \rightarrow $\neg\text{deformations}(\text{crystal},\text{sample})$

Fact: $\neg(\neg\text{deformations}(\text{crystal},\text{sample}))$

By applying the MTT rule to the fact and rule it is possible to achieve a further transformation ie

Rule: etched(crystal) \rightarrow $\neg\text{deformations}(\text{crystal},\text{sample})$

Rule: $\neg\text{etched}(\text{crystal})\leftarrow \neg(\neg\text{deformations}(\text{crystal},\text{sample}))$

This new state now matches the query and by applying the MPP rule it can be proved that the crystal that had a sample

with large deformation was not etched. Unfortunately, there are a number of difficulties with this approach to problem solving, especially when it is essential that the inference engine of an expert system operates automatically.

The first difficulty is concerned with the nature of logical inference. In many respects, the reasoning process of an expert is not based on logical inference, but common sense reasoning. The problem of the former system is that once it is proved that a certain condition is true, then it cannot be made untrue. This is termed monotonic reasoning. Common sense reasoning operates in a different way, and forms what could be called provisional truths or beliefs about a situation. These beliefs are held to be true until such time that new conclusion are required. This is referred to as non-monotonic reasoning.

A second difficulty with the application of rules of inference is knowing which rules to apply and when. It is easy for use to see from the above example which rules to apply to the problem because the methods used in formal proof involve an inherent problem solving capacity. When trying to automate predicate calculus, it is clear that the number of possible alternative applications of a small set of inferencing rules to even a simple problem are very large. The unguided application of rules of inference to a problem is likely to lead to a combinatorial explosion. The key issue is thus the development of knowledgeable proof procedures.

4.3.1.1 Non-monotonic Logic

There are a number of different types of logic available that overcome the problems of monotonic reasoning associated with predicate logic. They formalise the aspects of reversible reasoning and allow the inferring of consistent formulas from a set premise. In these types of logic simultaneous inconsistencies are recognised as invalid and any conclusions withdrawn in the event of new information disproving a previously consistent conjecture. Thus, it would not be permissible to have the crystal as etched and unetched at the same time. This would be regarded as inconsistent and the conclusions about it withdrawn. The operating conditions for non-monotonic logic have been specified by McDermott and others and can be summarised as follows (McDermott and Doyle 1980):

a) The system may allow mutually inconsistent sets of inferred formulas to exist, but that the order of the application of those formulae will each constitute a different set of premises. For example, the crystal is unetched, an etching solution is applied, the crystal is etched is a consistent set of premises on which conclusions can be based, but different from a set with a different order. Blocking of inconsistent inferences would be based on a definition of orders such that 'crystal is etched' would block the inference that 'the crystal is unetched'.

b) To overcome the loss of the property of iteration, due possibly to the withdrawal of previous conclusions, a fixed

set of stable formulae protected from additional inferences must be achieved .

c) To avoid circularity in non-monotonic reasoning, a system must have some way of dynamically altering the 'inferability' of the inferencing rules. In this way the continuous application and then withdrawal of the same inference will be avoided. This will permit the system to verify an inference if the assertion about to be inferred is inconsistent with all other inferences that have previously been made by the system from the current set of premises.

d) There are considered to be two differing and broadly based principles upon which the reversibility of logic can be based and these must be considered when developing a inferencing system based on non-monotonic logic. Firstly, that reasoning is reversible because it is uncertain and often due to conjecture ie 'Usually, objects of type X have attributes A. If B is an object of type X, then it can be deduced that B probably has the attributes A.' In this example we might believe that B has attributes A, but additional information could lead to a different conclusion if it contradicts the current situation. Secondly, that reasoning is reversible because it is introspective. This could be seen as the fact that reversibility is the result of certainty being based on the level of knowledge about a situation. The level of knowledge can change with time and, therefore, conclusions revised ie 'From the current state of knowledge K about X, it can be deduced that X has the

attributes A, but in the event of new knowledge N it cannot be deduced that X has the attributes A'.

By incorporating the four guide-lines into a logic system, it should be possible to develop an inferencing method that allows a proof to be reworked when evidence indicates that this is necessary. Three useful forms of logic have been formulated with the aid of these guide-lines and they are:

- a) Default logic (Reiter 1980)
- b) Non-monotonic Modal Logic (McDermott 1982)
- c) Autoepistemic Logic (Kleene 1977)

Default Logic

This is method of reasoning with incompletely defined worlds. The method is especially appropriate for dealing with expert system tasks since it is unlikely that all the information required for solving a domain problem will be available at run time. Default logic aims at completing our belief system by the most plausible conjecture. The process of applying default reasoning is based upon the application of patterns of inference in the form: 'in the absence of information to the contrary assume ...'.

Non-monotonic Modal Logic

The aim of this logic system is to overcome the problems of circularity associated with non-monotonic reasoning, only allowing the inferencing of consistent assertions. Broadly

speaking, the modal system is based on an axiomatic model of logic. The system strives to combine axioms, which are always true, by manipulating symbolic strings to produce new symbolic strings such that the complete axiomatic system approaches truth.

Autoepistemic Logic

This logic system is referred to as the logic of knowledge. It aims at modelling the beliefs of wholly rational agents who are capable of analysing their own beliefs (Moore 1985). It is capable of expressing statements such as 'if as agent one cannot believe p , then q is true'. Autoepistemic logic deals with the introspective aspects of thought that suggest that as a rational agent only logical consequences can be inferred from a set of beliefs, which in themselves are not necessarily true, and that during evaluation all the logical consequences must be taken into consideration with both the positive and negative aspects of introspection assessed. This means that the agent of the system must have complete understanding of the logical consequences of what is believed and what is not believed.

4.3.1.2 Automation of Proof

The development of non-monotonic logics goes somewhat towards modelling the common sense reasoning that is associated with human cognitive processes. The inherent weakness of the monotonic properties associated with predicate logic can be overcome by the instigation of default reasoning, modal logic (non-monotonic) and autoepistemic logic. The problems

of the automation of proof is the second weakness of logical systems. This issue centres on knowing which inferencing rules to apply and when to apply them in order to match items in a knowledge base using proof.

Resolution is the method for theorem proving that reduces a complex proposition into a solvable entity through the application of a single inference rule. This process underlies the functioning of the programming language PROLOG which it is believed provides a general solution to the automation of reasoning (Stalnaker 1988). However, in order to apply the resolution procedure, the knowledge stored as a binary predicate must be simplified into its clausal form. The simplification process requires that the predicate must be expressed as a list of 'or' connectives, which can be achieved through transformation. This is termed the disjunctive normal form. There is an algebraic approach to the transformation of predicate logic into equivalent representations, and these can be expressed as a series of laws (Chang and Lee 1973). Thus if X and Y are formulae:

a) $(X \wedge X) = X = (X \vee X)$	Idempotence
b) $(X \wedge Y) = (Y \wedge X)$	Commutativity
c) $(X \vee Y) = (Y \vee X)$	Commutativity
d) $((X \wedge Y) \wedge Z) = (X \wedge (Y \wedge Z))$	Associativity
e) $((X \vee Y) \vee Z) = (X \vee (Y \vee Z))$	Associativity
f) $((X \wedge Y) \vee Z) = ((X \vee Z) \wedge (Y \vee Z))$	Distributivity
g) $((X \vee Y) \wedge Z) = ((X \wedge Z) \vee (Y \wedge Z))$	Distributivity
h) $(X \vee \neg X) = T$	Complementarity
i) $(X \wedge \neg X) = F$	Complementarity

j) $\neg\neg X = X$	Involution
k) $(X \rightarrow Y) = (\neg X \vee Y)$	Duality Principle
l) $\neg(X \wedge Y) = (\neg X \vee \neg Y)$	The De Morgan law
m) $\neg(X \vee Y) = (\neg X \wedge \neg Y)$	The De Morgan law
n) $(X = Y) = (X \rightarrow Y) \wedge (Y \rightarrow X)$	Normalisation

Figure 4.6 Laws of Re-expression for all Formulae.

Through the application of these laws in matching sequence, the disjunctive normal form of predicate logic can be achieved automatically. This is a simple process for uncomplex expressions, but not sufficient for complex expressions.

With complex expressions the proposition is likely to contain quantifiers, and under these circumstances it is necessary to eliminate quantification by the transformation of predicate logic into what is called the Skolem form before it can be resolved. However, in order to perform this operation several stages are required all of which can be performed automatically within the inferencing mechanism (Kleene 1977):

- a) Transform formula into a prenex form by an algorithm
- b) Transform the matrix of this form by algorithmic means
- c) Apply the Skolem procedure to closed prenex form

4.3.1.3 Resolution

Through the implementation of the three procedures referenced above, binary predicates can be converted into clausal form using the application of general algorithms. Having achieved this, it is then possible to resolve the formulae by cancelling out different clauses when they are negated in one clause and unnegated in another. More specifically, this is the process of unification and can be represented as the co-occurrence of two clauses in the same set leading to the literals of the clauses being grounded in the same instances (Bell and Machover 1977) i.e.:

$$C1 = \{l1, \dots\} \text{ and } C2 = \{\neg l2, \dots\}$$

where:

l = literals

C = Clauses

R = Predicate Clause

and:

$$R' = (C1' \setminus \{l'\}) \cup (C2' \setminus \{\neg l'\})$$

R' is the resolutes of the instances of $C1$ and $C2$ which results in the cancellation of matching clauses in the set to which they belong. The resolution process requires that the complete set is analysed until its viability is established. A resolution algorithm exists to establish

whether a formula can be resolved, and is typical of the automation techniques behind logic programming:

While $F \in S$

 Select l_1, l_2, s_1, s_2 such that:

s_1 and s_2 are clauses ;

$l_1 \in s_1$ and $\neg l_2 \in s_2$ and l_1 and l_2 unifiable;

 Compute the resolute clause r ;

 Replace S by $S \cup \{r\}$;

End.

4.3.1.4 Reasoning

The resolution procedure provides the underlying mechanism for the reasoning process, and this allows logic programming to be used to automatically solve the problems within a domain using an exact methodology. Two forms of reasoning are commonly used, forward reasoning from the start state to the goal state and backward reasoning from the goal state to the start state (see Section 3.5). The formulae representing the assertions of the domain are usually separated into two categories: rules and facts. The rules express the general knowledge about the subject area and are normally stored in the knowledge base of an expert system architecture. They are constructed as implicational statements that state that 'given certain conditions the following actions will result'. The facts are also assertions, but are not implicational. The facts represent the current state of the problem and are normally stored in the database of an expert system architecture. To start the system working on a

problem a goal is defined and this presented to the system as a query. The task of the inference engine is then to prove the goal formula. To illustrate the way in which logic operates during this process consider the following set of assertions within the X-ray crystallography domain:

Rule 1

if Y is a sample of crystal X and if W is an output of crystal Z with $Z = X$ then Y can be matched to W.

$(\text{Sample}(X,Y) \wedge \text{Output}(Z,W) \wedge \neg \text{Equ}(X,Z)) \rightarrow \text{Match}(Y,W).$

Fact 1

Experimental curve is a sample of a test crystal configuration.

$\text{Sample}(\text{Test}, \text{Expt_curve})$

Fact 2

Simulation curve is an output of a simulation crystal configuration.

$\text{Output}(\text{Sim}, \text{Sim_curve})$

In this sample of expertise, there are three assertions. The extracted assertions are shown in two forms, with the top form expressing the common sense interpretation from the

domain, and the second form showing the predicate calculus structure for the same knowledge. The rule states that if a specified sample is from the test data produced by a specified crystal of known layers and thickness, and if a specified simulated output of a crystal of known layer and thickness is produced, then the experimental data can be matched to the simulated, but on the proviso that the sample data and the output data are from different sources. The first fact states that an experimental curve has been created from a specific test crystal. The second fact states that a simulated curve has been produced as an output from a simulated curve of a specific crystal configuration.

Having established the current status of the system and its relevant rule(s), a goal can be set, normally in the form of a query which has then to be proved by the system. A simple query in this situation could be whether or not the sample can be matched to the output of the simulation program. In predicate logic this would be represented as:

Query 1

Match(Expt_curve, Sim_curve)

Either of two search strategies can now be used to establish whether the query can be proved given the current facts available and the current knowledge stored in the system. As mentioned before the two methods are forward deductive reasoning and backward deductive reasoning.

Forward Deduction

In the forward deductive process, the deductive rules are applied to facts and rules in order to produce new knowledge, and the reasoning process terminates when the goal formula is found. Thus, given the assertions of the above example, the theorem that requires proof is:

$$(\text{Fact}(1) \wedge \text{Fact}(2) \wedge \text{Rule}(1) \rightarrow \text{Goal}(1),$$

And using the resolution procedure, the inferencing mechanism produces a new rule (Rule(2)) from Fact(1) and Rule(1) and so on:

Fact(1) Rule(1)
Rule(2)

Given that the rules and facts of the domain have been converted into clausal form, the example can now be proved using resolution:

Stage 1

Rule(1) and Fact(1) \rightarrow Rule(2)

Fact(1)

Sample(Test, Expt_curve)

Rule(1)

$\neg \text{Sample}(X, Y) \vee \neg \text{Output}(Z, W) \vee \text{Equ}(X, Z) \vee \text{Match}(Y, W).$

Rule(2)

$\neg \text{Output}(Z, W) \vee \text{Equ}(\text{Test}, Z) \vee \text{Match}(\text{Expt_curve}, W)$

Stage 2

Fact(2) and Rule(2) \rightarrow Fact(3)

Fact(2)

$\text{Output}(\text{Sim}, \text{Sim_curve})$

Rule(2)

$\neg \text{Output}(Z, W) \vee \text{Equ}(\text{Test}, Z) \vee \text{Match}(\text{Expt_curve}, W)$

Fact(3)

$\text{Match}(\text{Expt_curve}, \text{Sim_curve}) \vee \text{Equ}(\text{Test}, \text{Sim}) = F$

Stage 3

Fact(3) now corresponds to goal(1) and can be proved by resolution through the negation of the goal.

Fact(3)

$\neg \text{Goal}(1)$

$\text{Match}(\text{Expt_curve}, \text{Sim_curve}) \quad \neg \text{Match}(\text{Expt_curve}, \text{Sim_curve})$

Each of these three stages shows how resolution is used in forward deduction to prove an initial goal.

Backward Deduction

In backward deduction, the deduction rules are applied to the goal and the rules in order to generate new sub-goals, and the reasoning halts once all the facts have been proved.

In terms of logic, the application of backward deduction can be said to represent F_1, \dots, F_n and G , where G is the logical consequence of F_1, \dots, F_n provided that $(\neg F_1, \dots, F_n \vee G)$ is also true. Using the same example, the proof follows three stages:

Stage 1

Goal(1) $\wedge \neg(\text{Rule}(1)) \rightarrow \text{Goal}(2)$

Goal(1)

Match(Expt_curve, Sim_curve)

$\neg(\text{Rule}(1))$

Sample(X, Y) \vee Output(Z, W) $\vee \neg\text{Equ}(X, Z) \vee \neg\text{Match}(Y, W)$.

Goal(2)

Sample(X, Expt_curve) \vee Output(Z, Sim_curve) $\vee \neg\text{Equ}(X, Z)$

Stage 2

Goal(2) $\wedge \neg(\text{Fact}(1)) \rightarrow \text{Goal}(3)$

Goal(2)

Sample(X, Expt_curve) \vee Output(Z, Sim_curve) $\vee \neg\text{Equ}(X, Z)$

$\neg(\text{Fact}(1))$

$\neg\text{Sample}(\text{Test}, \text{Expt_curve})$

Goal(3)

Output(Z, Sim_curve) $\vee \neg\text{Equ}(\text{Test}, Z)$

Stage 3

Goal(3) $\wedge \neg(\text{Fact}(2)) \rightarrow T$

Goal(3)

Output(Z, Sim_curve) $\vee \neg\text{Equ}(\text{Test}, Z)$

$\neg(\text{Fact}(2))$

$\neg\text{Output}(\text{Sim}, \text{Sim_curve})$

T

$\neg\text{Equ}(\text{Test}, \text{Sim}) = \text{True}$

The three stages of backward reasoning automatically proves the query and again the problem solved, and in both reasoning directions the results of inferencing permits the matching of the experimental curve with the simulated curve.

4.3.2 Statistical Reasoning

There are been a growing interest in the use of numerical methods of reasoning which are in sharp contrast to the approach taken the system of logic outlined in the previous section. In regard to logical reasoning, the manipulation of symbols was thought to be the central requirement of an intelligent system, with numbers being defined as just another intermediate step of the thinking process, and of little interest to the A.I. community. However, the development of expert system technology has focused attention on the need to deal with uncertainty, and has

utilised the numerically based theories on probability and belief functions (Shafer 1976). In a practical sense, inputs to expert systems and indeed the internal representations of the system may be numerical, especially when dealing with scientific domains of expertise. What is more, intelligence can be defined at many different levels, some more appropriate to understanding than others. This is a general philosophical point, reiterated by Marr in his statement that " It's no use, for example, trying to understand the fast Fourier transform in terms of resistors as it runs on an IBM 370 " (Marr 1982). This position suggests that thought (inference) may sometimes be non-symbolic and numeric in character depending on the nature of the problem solving process, and that a cognitively based expert system should, thereby, operate both symbolically and numerically. There are a number of methods used to frame a problem around statistical techniques, and the most generally accepted way of reasoning is with probability. Bayes' theorem is the classical method for reasoning through an expert system, and there are a number of examples of systems that employ this approach, including an adaption of the technique used on the MYCIN project (Buchanan and Shortliffe 1984), and the implementation of subjective Bayesian rules operating within PROSPECTOR as an inference network (Duda, Gashnig and Hart 1979). Fuzzy reasoning is an alternative method for statistical inference, and is an extension of Boolean logic to real numbers. The method was designed by Zadeh (1965), and has been implemented in Cadiag-2, a diagnostic expert system (Adlassnig and Kolarz 1982). Finally, a system for

manipulating degrees of belief has been formulated by Shafer from the earlier work of Dempster (Shafer 1987). Its application to expert system technology is discussed by Gordon and Shortliffe (1985).

4.3.2.1 Bayesian Logic

Originally devised by Thomas Bayes, Bayesian logic is a method of expressing the uncertainty of a hypothesis (H) given the evidence (E) available. This is particularly important to inference as it is a way of saying how certain we are about a particular inference. In predicate logic, it becomes possible to say that given fact (P) and the deduction (Q) assuming (P thereby Q) it is possible to assign a probability to P expressing the likelihood of it being true.

As an illustration of the model, an example of the X-ray crystallography domain shows how a probability factor for interference fringes could be assign to an identified peak of an X-ray Rocking Curve for a given crystal structure. In terms of understanding the consequences in the domain, the more evidence for interference, the greater the probability of interference and the less likely it is that the selected peak reflects the true structure of the identified crystal layer or substrate.

Under these circumstances, it is possible to determine the probability that the peak contains interference from the evidence, so for example:

H = Peak(X) contains Interference.

E = Layer(Y) is less than 0.5 microns.

Form the statistical model three probabilities are now required:

- a) $P(H)$: the probability that selected peak contains interference.
- b) $P(E|H)$: probability that the layer is less than 0.5 microns, assuming that interference exists.
- c) $P(E|-H)$: probability that the layer is less than 0.5 microns, assuming there is no interference

The probability that the peak contains interference from a layer of less than 0.5 microns can be deduced from the known probabilities of the model and calculated from Bayes' rule:

where

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

and

$$P(E) = P(E|H)P(H) + P(E|-H)P(-H).$$

The rule states that the probability of the peak containing interference if the identified layer is less than 0.5 microns is equal to the ratio of the probability that the peak has interference and is less than 0.5 microns, over the

probability that the layer is less than 0.5 microns whether there is interference is not.

If the experience of the expert is used to determine the knowledge of the model it is possible to calculate $P(H|E)$. In this instance the domain knowledge is as follows: $P(H) = 0.45$, $P(E|H) = 0.60$, $P(E|\sim H) = 0.11$

thus

$$P(E) = (0.60 * 0.45) + (0.11P * (1 - 0.45))$$

$$P(E) = 0.3305$$

and

$$P(H|E) = \frac{0.60 * 0.45}{0.3305}$$

$$P(H|E) = 0.817$$

The model predicts that the probability of interference to a peak when the selected crystal layer is less than 0.5 microns is approximately 0.817. The model can also be used to determine the probability of interference if the selected layer is greater than 0.5 microns as follows:

$$P(H|\sim E) = \frac{P(\sim E|H)P(H)}{P(\sim E)}$$

$$P(H|\neg E) = \frac{(1 - 0.6) * 0.45}{(1 - 0.3305)}$$

$$P(H|\neg E) = 0.2689$$

In comparing these two results it can be assumed that from the three standard probabilities expressed by the expert, the probability of interference occurring in the selected peak of an X-ray rocking curve is four times more likely to occur when the layer is less than 0.5 microns.

The Odds Rule

Bayes rule can be expressed in odds form and illustrates the way in which the odds of an hypothesis change as evidence is formulated. Odds is simply the probability of the outcome for the hypothesis over one minus the probability of the outcome. This is achieved by dividing the formula for the probability of H given the evidence (H|E) by the negation of H given the evidence ($\neg H|E$) thus:

$$\frac{P(H|E)}{P(\neg H|E)} = \frac{P(E|H)P(H)}{P(E)} \cdot \frac{P(E)}{P(E|\neg H)P(\neg H)}$$

and

$$\text{odds (O)} = \frac{\text{probability (P)}}{1 - \text{probability (P)}}$$

therefore

$$O(H|E) = \frac{P(E|H)}{P(E|\neg H)} \cdot O(H)$$

In this form $O(H)$ represents the prior odds of the hypothesis and $P(E|H)/P(E|\neg H)$ is the likelihood ratio. The higher the ratio (> 1) the more likely it is that in the presence of the evidence (E) the hypothesis will be true. This ratio is referred to as the sufficiency factor (SF) thus:

$$O(H|E) = SF \cdot O(H)$$

Conversely, if the evidence is not true then it follows that a necessity factor can be achieved in the following manner:

$$\frac{P(\neg E|H)}{P(\neg E|\neg H)} = \frac{1 - P(E|H)}{1 - P(E|\neg H)}$$

$$O(H|\neg E) = \frac{P(\neg E|H)}{P(\neg E|\neg H)} \cdot O(H)$$

Again $O(H)$ is the prior odds of the hypothesis, but the odds of the hypothesis being true in absence of evidence ($O(H|\neg E)$) is dependant on the likelihood ratio approaching 0. Under these circumstances the presence of E reduces the likelihood of H and is said to be sufficient for $\neg H$, whilst

E is necessary for H since the absence of E ($\neg E$) results in H being unlikely. Thus a necessity factor (NF) is created:

$$O(H|\neg E) = NF \cdot O(H)$$

The odds form of the Bayesian rule can now be applied to the previous example from the X-ray crystallography domain for both the SF and NF, allowing the computation of the odds of interference to the peak given the presence or absence of evidence (the thickness of the layer/substrate). As before, the domain knowledge is as follows: $P(H) = 0.45$, $P(E|H) = 0.60$, $P(E|\neg H) = 0.11$.

The prior odds that the peak contains interference is:

$$O(H) = \frac{P}{1 - P} = \frac{0.45}{1 - 0.45} = 0.818$$

The necessity and sufficiency can be calculated:

$$SF = \frac{P(E|H)}{P(E|\neg H)} = \frac{0.60}{0.11} = 5.455$$

$$NF = \frac{1 - P(E|H)}{1 - P(E|\neg H)} = \frac{0.40}{0.89} = 0.449$$

and combined with prior odds to give both the odds that peak has interference when the layer is less than 0.5 microns and when the layer is greater than 0.5 microns.

$$O(H|E) = SF.O(H) = 5.455 * 0.818 = 4.462$$

$$O(H|\neg E) = NF.O(H) = 0.449 * 0.818 = 0.367$$

This is an alternative expression of Bayesian logic which is useful when using this method for statistical inference.

Inferencing with Bayesian Logic

Bayesian logic may be used to suggest a hypothesis and maintain its credibility through the application of necessity and sufficiency co-efficients. The formulation of the hypothesis can be structured as an inferencing network and Figure 4.7 is an example of an arc from an inferencing network for the X-ray crystallography domain.

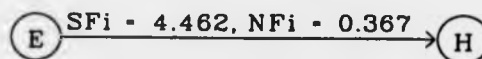


Figure 4.7 Inferencing Arc for Interference on Peak.

By applying the two probability equations it is possible to strengthen or weaken the hypothesis by multiplying the existing odds with the newly calculated odds

$$O(H|E) = SF.O(H)$$

$$O(H|\neg E) = NF.O(H)$$

However, in most expert system problems the certainty of the evidence may not be 100 percent, and in these cases it is necessary to take account of this uncertainty. In the present scheme there are two extreme values lying at 0 and 1 which is that the evidence is true or the evidence is false respectively. Uncertainty of the evidence must lie somewhere between these two values. For example, in growing a crystal, the expert may not be certain of the thickness of the layer. If the growth process produces an 60 percent confidence level for layers in the region of 0.5 microns, then it can be said that the evidence contributing towards the hypothesis (E^{\wedge}) has a probability of 0.6. This can be expressed as $P(E|E^{\wedge})$ and combined with the two extremes to form a linear interpolation of $P(H|E^{\wedge})$ thus

$$P(H|E^{\wedge}) = P(E|E^{\wedge}).P(H|E) + (1-P(E|E^{\wedge})).P(H|\neg E)$$

If observations show that the probability of the layer thickness E being known is 0.6 then the updated probability given uncertain evidence is

$$P(H|E^{\wedge}) = 0.6 * 0.817 + (1 - 0.6) * 0.2689 = 0.598$$

This is a linear effect that can be used to predict the probability as the possibility of evidence changes. However, the relationship between evidence and hypothesis might not necessarily be linear and both PROSPECTOR and MYCIN have used methods for overcoming inconsistencies in this relationship through the identification of "zones of

inconsistency" (Shortliffe 1976). This method typically involves the use of a function to apply different probability equations depending on the trends in the evidence; is the evidence weakening or strengthening.

Multiple sources of evidence bearing on a hypothesis can be computed by multiplying their independent sufficiency and necessity factors to give overall sufficiency.

Thus:

$$SFI = SFi1.SFi2.SFi3.SFin$$

$$NFI = NFi1.NFi2.NFi3.NFin$$

This can then be used to calculate the odds for the hypothesis from multiple sources of evidence. Again, as with single sources of evidence the certainty of the evidence can be adjusted given $P(E|E^{\sim})$.

Figure 4.8 is a diagram of an inferencing net with multiple arcs.

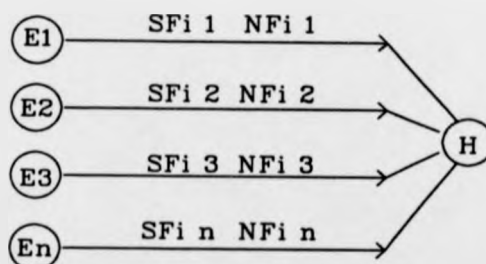


Figure 4.8 Inferencing Net with Multiple Arcs

4.3.2.2 Fuzzy Logic

Fuzzy logic was designed by Zadeh as a means of applying Boolean logic to real numbers. In the field of expert systems, this method can be used both to represent knowledge and make inferences from it. Sometimes referred to as subjective bayesian logic, or possibility theory. This technique can be applied to the inferencing rules of propositional logic to produce a system of possibilistic logic. This is an alternative method to the use of pure bayesian reasoning since it takes account of the problems associated with the modelling of the former system. For example, it is not always possible for an expert to know the probabilities of all events given the evidence available. In the domain of X-ray rocking curve analysis, the expert may not have the data available to calculate the probability of interference to a selected peak of an X-ray spectrum. It is also unlikely that accurate figures will be accessible on the probability that layer thickness in a crystal can be correctly identified as $<$ or $>$ 0.5 microns given the absence or occurrence of interference in the corresponding peak of the X-ray rocking curve.

Set Membership

Set theory is used in fuzzy logic to describe the strength of membership of a fuzzy set of objects (F) to a known set of objects (U). This is denoted by the grade membership of an object of subset F within U, given the membership function $UP(u)$ for all elements $u \in U$. The closer the subset F is to 1.0, the stronger the grade membership of u in F.

This means that an object can "possibly" be a member of a set. In a normal set the value of a object can only be 1 or 0, denoting membership or non-membership to the set. The sums of probabilities for selecting all the objects of a normal set must also always equal 1. In a fuzzy set the possibility of selecting all the objects within it does not have to equal 1 (Zadeh 1978). For example, the structure of a simple rocking curve with one peak could be classified as Substrate, Single Layered, or MQW. The probability that any rocking curve could be one of these classifications might be 0.5, 0.4, and 0.1 respectively. However, the possibility that any simple rocking curve could be one of these classifications might be 0.7, 0.8, or 0.4 respectively. This fuzziness also holds for the matching of sets and the descriptions of empty sets. Fuzzy sets F and G are only equal if $U_F(y) = U_G(y)$ for all elements $y \in U$, and a set is only empty if all elements of the membership function $U_x(y)$ equal 0. These rules of fuzziness have been extended to cover intersection, union, and difference between sets, and the compositional rules of logic: disjunction, conjunction and implication (Zimmerman 1987).

Inferencing with Fuzzy Logic

The application of fuzzy logic to inferencing allows non-crisp rules of inference. Using the standard modus ponens rule ($A \rightarrow B, A \vdash B$), inexact matching can be formed such that:

Premise: The crystal is almost etched
 Implication: IF the crystal is etched THEN test
 Conclusion: almost test

This type of logic is an extension of the standard rules of inference used in non-fuzzy techniques, and has been applied successfully to a number expert systems. These include Sphinx, a system for general medical diagnosis (Fieschi 1982), and Cadiag-2 for a limited selection of dysfunctional diseases (Adlassnig and Kolarz 1982).

4.3.2.3 Dempster Schafer Calculus

Dempster Schafer Calculus outlines a framework for combining the strength of a piece of evidence in relation to a set of defined hypotheses. The approach is a development of Bayes' rule, because it defines the amount of certainty attached to a piece of evidence. The theory expresses the degree of belief in a piece of evidence, and assumes that it contributes to both the belief and disbelief in a hypothesis (Shafer 1987).

The frame of discernment (Θ), is the key concept of the theory, and represents a world of mutually exclusive events, equivalent to the sample space (τ) of probability theory (De Finette 1976). In this respect, probability theory and Dempster Schafer Calculus map events in the same way. However, Dempster Schafer Calculus defines the number of possible hypothesis as the power of number of combinations of events, or 2^{Θ} . In probability theory this is simply τ . For example, if the number of events required to describe an unknown rocking are: multi-quantum well (MWQ); substrate only (SO); and single graded layer (SGL); then the number of events in τ is 3, whereas the number of events in

\mathbb{E} is 8. \mathbb{E} accounts for all the possible subset combinations of evidence in favour and evidence against.

Given that A is a subset of \mathbb{E} , then $p(A)$ is the function that describes probability of the event occurring, and $m(A)$ the function that describes the portion of total belief assigned to A . From this there are two important assignments that mathematically describes events in \mathbb{E} . The first is the basic probability assignment which maps the power set of \mathbb{E} to numbers ranging between 0 and 1. Two conditions are required to satisfy a probability assignment:

- (a) The probability of a null event is 0;
- (b) The sum of probability numbers of all subsets is 1.

The second assignment is the degree of belief in power set of \mathbb{E} . For subset A this is $Bel(A)$. Three conditions need to be satisfied for a belief assign:

- (a) The belief in a null hypothesis is 0;
- (b) The belief in \mathbb{E} is 1;
- (c) The sum of beliefs of A and $\neg A$ must be ≤ 1 .

Plausibility ($Pl(A)$) can be defined as the degree of belief which can be attached to an event that is believed ($1 - Bel(\neg A)$). This is the maximum amount of belief that can be assigned to an event. $Pl(A)$ is, therefore, the upper limit of probability and $Bel(A)$ the lower limit of probability in a function $p(A)$.

Propagation of Belief

When there is more than one probability assignment for \mathbb{E} , rules of combination are required to calculate the amount of overall belief in each hypothesis given the degree of belief and disbelief in each assignment. The rules of combination express the weight of conflict between different beliefs, and enable the Propagation of probability according to the rules of Dempster Schafer Calculus. The weight of conflict is called K , and if two beliefs ($Bel1$ and $Bel2$) are in total conflict then $K=0$, and if they are in total agreement then $K=1$. Any combinations in between can be calculated according to the orthogonal sum:

$$m1 + m2(A) = K \sum_{X \cap Y = A} m1(X) \times m2(Y),$$

where $m1$ and $m2$ are two probability assignments to \mathbb{E} , and X and Y are the elements of the assignments. Table 4.3 gives an example of a set of probabilities for combination into a set of beliefs given two probability assignments. These are related to an unknown rocking curve which might be classified as belonging to any of three types: MWQ; SO; or SGL. The first assignment ($m1$) provides strong evidence that the rocking curve is a SO (0.9). However, the second assignment ($m2$) provides good evidence that it could be a SGL (0.6), and weaker evidence that it either MQW or SO (0.15). By multiplying the matrix of probability values for \mathbb{E} , the probabilities of combined subsets can be found.

	M2	SGL(0.6)	MQW,SO(0.15)	E(0.25)
M1				
MQW (0.9)		$\theta(0.54)$	(MQW) (0.135)	(MQW) (0.225)
E (0.1)		SLG(0.060)	(MQW,SO) (0.015)	E(0.025)

Table 4.3 The Rules of Combination for Probability Functions

The separate beliefs of the two probability assignments can be combined, according to Gordon and Shortcliff (1985), to give a propagated belief measure for an overall assessment of belief in the evidence. From Table 4.3 this is as follows:

$$K = 1/1 - E\theta = 1/1 - 0.54$$

$$m1 + m2(|MQW|) = (0.135 + 0.225)/1 - 0.54 = 0.783$$

$$m1 + m2(|SLG|) = 0.060/1 - 0.54 = 0.130$$

$$m1 + m2(|MQW,SO|) = 0.015/1 - 0.54 = 0.033$$

$$m1 + m2(E) = 0.025/1 - 0.54 = 0.054$$

In the worked example from Table 4.3 the strongest belief in the evidence is for the MQW structure. This has the best

probability of being correct given the two sets of probability assignments.

4.3.3 Discussion

Symbolic inference has been used extensively within expert systems. The simplest form of inference uses modus ponens in a forward chaining action (see Section 4.3.1). This is a simple system to operate, but has little reasoning power. Backward chaining provers add to the power of such systems, but neither are defined enough to solve all problems. Resolution provides a more general logical proof and forms the basis of the logic programming approach of the PROLOG language. However, these logical systems do not allow the withdrawal of facts if evidence supporting them is contradicted, and require restricted forms of representation. Furthermore, uncertainty often exists in knowledge, sometimes information may be missing, and sometimes the knowledge may be expressed numerically. In these situations monotonic reasoning is inadequate. Non-monotonic reasoning systems tackle these issues by introducing default reasoning to express certainty (Moore 1985), and dependency backtracking to allow the withdrawal of disproved facts (Stallman and Sussman 1977). A non-monotonic reasoning system maintains consistency by backtracking every time an inconsistency is found. The experience as such is that its reasoning power is greater than monotonic systems, but at the price of memory and processing time. The main difficulty is that all paths of reason have to be maintained to enable the withdrawal of

past evidence. Monotonic systems do not have to keep track of reasoning because once a proof is found it never has to be re-evaluated. In this respect, non-monotonic systems can suffer from Propagation fatigue. Statistical reasoning systems use numbers to represent the results of inference. For example, the probability of an event occurring, the strength of a piece of evidence, the possibility of a hypothesis being correct, and so on. Statistical reasoning is, therefore, good at representing uncertainty. Conditional probability, or Bayesian logic is the most widely used way of statistically representing uncertainty. It has a robust methodology, and has been used in many expert systems (Agogino and Rege 1987, Heckerman, Horvitz and Nathwani 1989). Fuzzy logics have been developed and applied to expert systems to deal with knowledge that imprecise. Reasoning can, thereby, be performed when predicates are not crisp. Inexact pattern matches can be performed using this logic, but at a computational price. Critics claim that the fuzzy logic methods can be just as well defined using probability theory (Cheeseman 1985). Dempster Shafer Calculus extends the use of probability theory by assigning degrees of belief in the evidence. This gives a more intuitive feel to the numbers produced by such a method. The problem is that belief is a power relation and the combinations used in the frame of discernment explode as the number of events increase. Another concern is that only an experimental system called Gertis has been built to exploit Dempster Shafer Calculus (Yen 1989), and currently no really effective procedures have been produced for drawing

inferences from belief functions. Experimental comparisons with the same rules and data have been performed using the three statistical techniques, and results favour the Bayesian approach (Wise and Henrion 1986). The conclusions of Heckerman (1988) are that the Bayesian scoring mechanism is the best.

4.4 Conclusions

The use of control with knowledge is the key element of an expert system core. Heuristics alone will always fail because if all solutions are required then all paths must be explored. Knowledge restricts search and, thereby, limits the size of the search space. However, search is necessary when knowledge runs short. The extent that either is used in an expert system depends on the application. R1 was required to find a satisfactory configuration to a computer set-up. Mycin was required to give the precise diagnosis of an infectious disease. The latter depended far more on knowledge than the former. As time has passed these divisions have grown. Each new expert system spawns a range of new techniques, and there are now as many techniques as expert systems, but with few guide-lines on their use. This means that there is no one emphasis on the design of an expert system. In implementing solutions to problems through the design of an expert system core, the picture is further complicated by the number of competing configurations available for representing knowledge and performing inferences. Indications are that Bayesian logic handles uncertainty well, and a degree of inferred logic is required

to maintain a database of facts. Mixed modes of representation are probably necessary because knowledge can be both structured and unstructured, certain and uncertain, local and global, critical and non-critical. This helps reduce the possibilities. However, there are so many combinations available it is difficult to know from which to choose. The question that arises from this is how do you match techniques to domains?

The next chapter introduces the domain of X-ray rocking curve analysis which has problem characteristics that are suited to an expert system approach. The nature of the domain will be explored and the results of the analysis used to suggest an overall structure for an expert system core. A structured design procedure for implementing existing A.I. techniques will be employed to address the issue of matching techniques and domains.

Chapter 5

A Prototype Expert System Shell for X-ray Rocking Curve Analysis

5.0 Introduction

This chapter will examine the design of an expert system core using existing techniques from the field of artificial Intelligence. The design will be motivated by the domain of X-ray rocking curve analysis, and the approach taken will be systematic, dividing the expert system architecture into broad needs, and matching each need to a suitable technique. The outcome of this matching process will be an integrated design from which a prototype expert system shell, or system without knowledge, will be built. This design methodology is different from most used in expert system research, since it tries to match domain and design requirements.

Part One (The Domain)

5.1 What is X-ray Rocking Curve analysis

The Rocking Curve Analysis is a superior technique for analysing crystal structure (Halliwell and Lyons 1984). It is most usefully applied to the analysis of semiconductor device structures used in micro-chip manufacture. When semiconductor crystals are grown, they are usually composite materials made of a substrate of one material upon which is grown other materials. Typically, the crystal grower must be able to identify the compositions of these superlattices along with a number of important parameters which include:

the thickness of layers grown on top of the substrate; the matching, misorientation, grading, and roughness between layers or between layers and the substrate; period thickness (which is the average potential well plus barrier thickness, defined in electrical terms); any defects between layers, and the radius of curvature of the crystal. These parameters are known as structural parameters and they provide a mechanism for the description of the superlattice. To describe these parameters, crystal growers can use double X-ray diffraction which generates an X-ray rocking curve. The optics consists of an X-ray source, usually made of copper, a first crystal that monochromates and collimates the source, and a sample which diffracts the conditioned beam to a detector (see Figure 5.1).

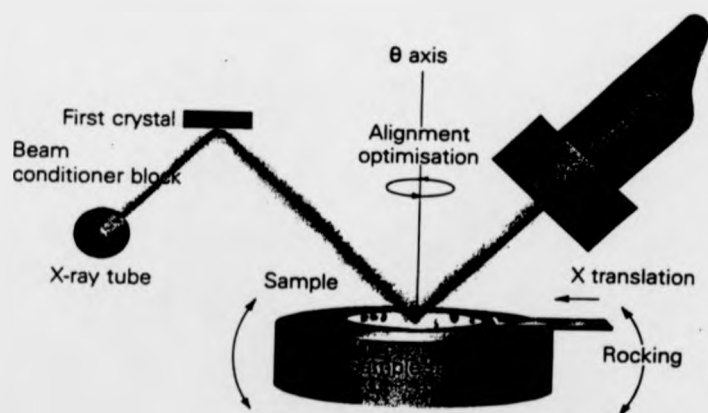


Figure 5.1 The Optics of Double X-ray Diffraction

The rocking curve is the spectrum obtained during the rotation of the second crystal through an angle at which it diffracts strongly to produce a spectrum that is highly sensitive to the structure of the outer layers of the sample. The output from the crystal can be used as a quality control index for advanced semiconductor materials.

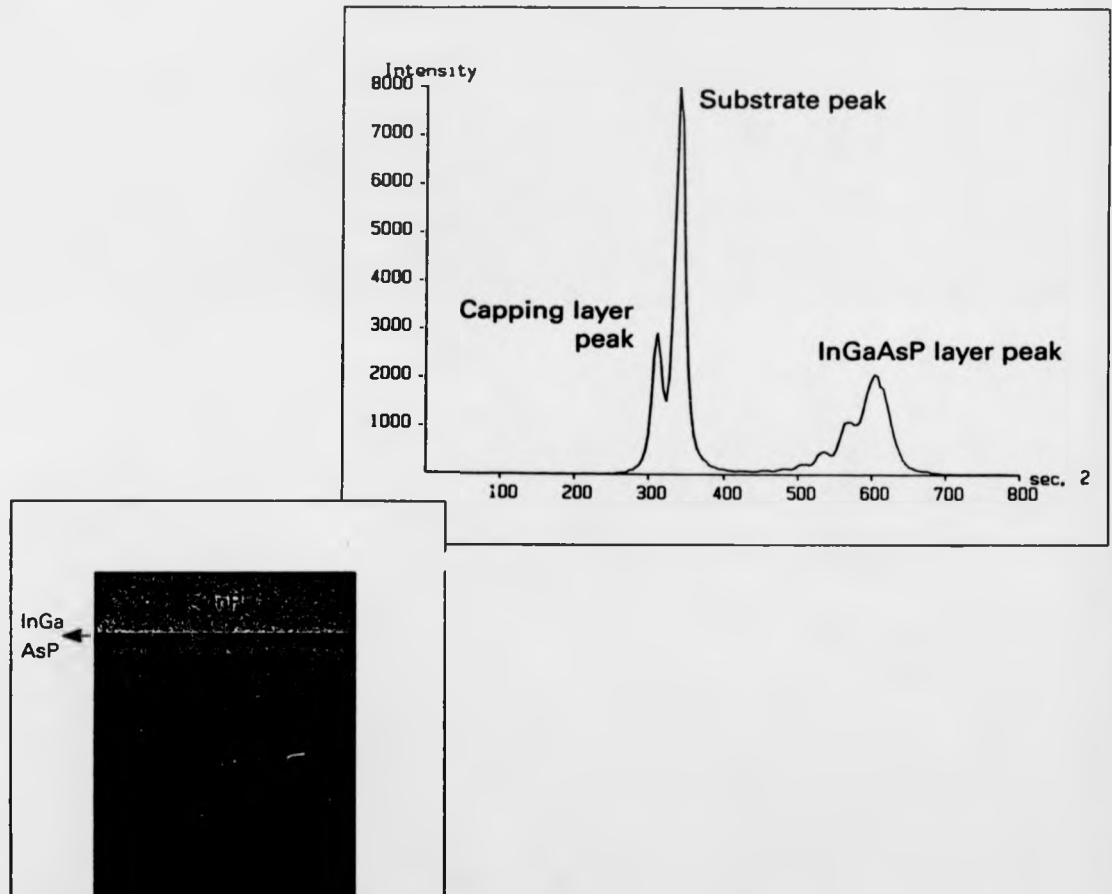


Figure 5.2 A Layered Structure and its Associated X-ray Rocking Curve

Figure 5.2 shows a typical rocking curve produced by double X-ray diffraction from a wafer and plotted using Bede Scientific Instrument's Double Crystal Control software.

5.1.1 Diffraction Theories

There are a number of detailed theories which model the way in which X-rays diffract from a crystal lattice. The simplest is kinematical theory, and this agrees well with experimental results of powder diffraction (Zachariasen 1945). Dynamical theory is more complex, and takes account of absorption effects of the crystal lattice, and, thereby, models larger crystal volumes. Finally, there are the Takagi Taupin equations, which have been formed into a more general theory of diffraction, and adapted for use in simulated models of X-ray diffraction.

Kinematical Theory

The manner in which X-rays diffract from the surface of the crystal obeys the fundamental relationship of X-ray diffraction - Braggs Law

$$A_{hkl} = \lambda / 2 \sin \theta$$

where:

A_{hkl} = then lattice parameter of the crystal

λ = the X-ray wavelength

hkl = the indices of the measured lattice planes

This is the kinematical theory, and predicts the condition at which maximum diffracted intensity occurs (Speriosu 1981). Figure 5.3 illustrates the phase relationship that results from the crystals' D spacing to give a changing rate of X-ray diffraction intensity for a given crystal rotation.

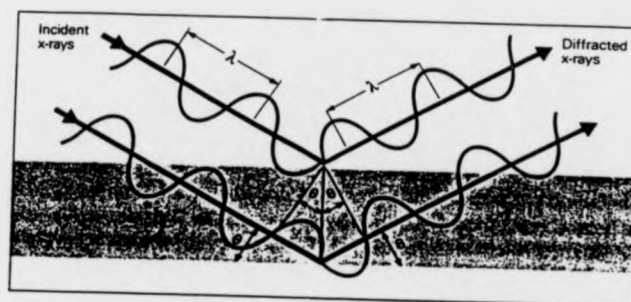


Figure 5.3 The Demonstration of Bragg's Law for X-ray Diffraction. (Copied from Philips Pamphlet ref 9498 700 10712)

Each crystal can be divided into unit cells that reflect the atomic structure of the lattice. The Bragg condition can be found by summing the contribution of each unit cell of the crystal lattice within the crystal volume for its given structural factor. The result is an intensity that is proportional to the both the volume of the crystal and the square of the structural factor. However, for the condition to work it is necessary to assume that each unit cell has the same exciting field, which is at a magnitude equal to the incident beam. What is more, in a near-perfect crystal

the diffracted beam is at the correct angle to rediffract off the back of the crystal at the original angle of diffraction, and this condition is not accounted for in kinematic theory.

These conditions are only met for small crystal volumes of an imperfect structure, but for large crystal volumes of near perfect structures, the kinematic theory becomes an inaccurate measure of X-ray diffraction. The type of structures investigated by X-ray rocking analysis tend to be of this type.

Dynamical Theory

Dynamical theory models the behaviour of X-rays when diffracted from large near perfect crystals (Halliwell, Lyons and Hill 1984). The theory takes account of absorption effects on wave fields created by the excitation of each unit cell in the crystal, and, hence, gives a more accurate measure of X-ray diffraction. According to Loxley: 'The wavefields excited by the incident beam are given by the points at which the inward facing surface normal cuts the dispersion surface (the so-called tie-points), the starting point of n being given by the deviation of the angle the incident beam makes with the diffracting planes from the exact Bragg angle, $(\Delta \theta)$.' Given these rules for dynamical theory he adds 'Now, the relative strengths of the direct and diffracted beams emerging from the crystal depend on the position of the tie-point selected and, hence, on the parameter $(\Delta \theta)$. Thus, as the crystal is rotated, the diffracted intensity changes, giving the rocking curve

its finite width. The important point about Bragg reflection, is that there is a range of ($\Delta\theta$) for which no tie-points are excited and hence no wavefields exist within the crystal. This is the range of total Bragg reflection' (Loxley 1987).

Generalised Diffraction Theory

A more general theory of diffraction was developed independently by Takagi (1962), and Taupin (1964) to describe the passage of X-rays through a crystal given any lattice distortion. The wavefield is the same as described by dynamical theory except that the incident and diffracted amplitudes are described as slowly varying functions of position.

These equations developed by Taupin and Takagi are solved within a simulation package called RADS (Bebe Scientific Instruments), and enables the user to mimic the experimental conditions for X-ray double crystal rocking curve analysis through input to a simulation program. As the Taupin and Takagi equations are mathematically rigorous, they accurately simulated the X-ray rocking curve plot produced using double X-ray diffraction equipment (see Figure 5.1).

5.1.2 Simulating a Rocking Curve

Simple crystal lattices give rise to simple rocking curves, and in these cases the structural parameters can be fairly easily read from the curves. This, though is not generally the case, especially in those circumstances where the layers

are very thin, for example, below 1pm, or there are many layers, or many repeating layers. In these cases, the experimental rocking curve is complex with interactions between peaks resulting in peak shifts, peak cancellations and interference effects. In such situations a model of the crystal lattice is necessary to deduce its structure.

Unfortunately, it is not possible to directly reconstruct the rocking curve because both the phase and intensity of the X-ray beam are required, and the former cannot be measured. There is, however, another method at the disposal of the expert for solving this problem. If the structure of a crystal is known, the expert can derive the rocking curve for that crystal. This is characteristic of many reversible problems in physics and general engineering. There is a well-defined method for doing this. This involves solving the Takagi-Taupin equations to give the rate of change of diffracted to incident beam amplitudes as a function of depth below the surface (Macrander, Minami and Berreman 1986). These calculations have been used to develop simulation software for double X-ray rocking curve analysis (Hill 1986). Whilst performing these calculations, allowances are made for structural parameters (see Appendix 1, pp3)

Having accounted for these factors a rocking curve can be simulated, so producing a model of the experimental rocking curve. When simulating a rocking curve in this way the structural parameters of the intended experimental structure are known. However, because the growth process is neither fully controllable or understood (Tjahjadi and Bowen 1989),

it is probable that when simulating a complex structure the experimental and simulated models will not match. It is under these cases that expertise is brought to bear upon the simulation process, iteratively altering the structural parameters of the simulated model until a satisfactory match is achieved. More explicitly, the problem now is one of describing the structure of a crystal in reasonably precise terms, producing a rocking curve for that crystal, and comparing it with the experimental rocking curve. The simulation algorithm is repeated until the derived rocking curves theoretically approaches the shape of the experimental rocking curve. When the match is very close, the expert can be confident that an adequate structural description of the superlattice crystal has been achieved. This method is what is meant by rocking curve analysis. It is not a trivial method, because the expert must first determine a structural description of the superlattice which is going to be simulated. This is achieved through analysis of the experimental rocking curve using the parameters outlined in Appendix 1. The problem is that it is very difficult to formalise the analysis of these curves. For example, a novice might perform 50 or more matching cycles between the simulated curve and experimental curve to achieve a result, whereas the expert may perform this in operation less than 10. One reason for this difference may be that the expert has more available information. However, even with the same information available the difference still exists. This suggests that the problem is essentially

a descriptive one, dependant on the shapes of the rocking curves themselves.

5.1.3 Descriptions of Rocking Curves

A profile of the material can be obtained by exposing a sample to X-rays in an X-ray diffractometer (Bede Scientific Instruments, 1987). Output is a 2D graphical representation of the composition and structure of a crystal lattice. The x-axis represents the angle through which the sample is rotated or rocked in an X-ray beam, and the y-axis the diffracted intensity of X-rays from the surface of the crystal. This creates a series of recognisable features reflecting the generating structure (see Figure 5.4).

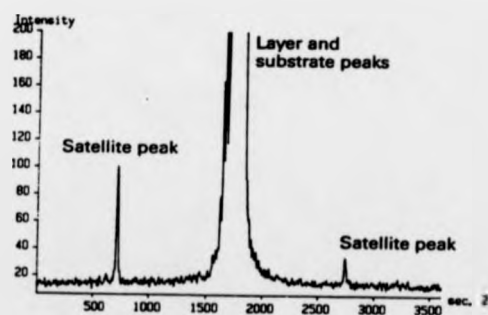


Figure 5.4 Typical Features in a Rocking Curve

There are an infinite number of semi-conductor materials available for analysis, and, consequently, the potential to produce an infinite number of rocking curves.

Some of these curves are simple and reflect the structure from which they originate, other are more complex and do not do have structural correspondence. Furthermore, differing structures converge to produce very similar rocking curve. To identify the rocking curve structure, therefore, it is necessary to know what the crystal grower intended to produce, and have a matching model for the structure using rocking curve simulation. Expertise helps in producing the matching model, by anticipating the characteristics of the rocking curve.

Descriptive types

There are a number of types of materials and corresponding rocking curve profiles. For example, substrate only, a single layer, and MQW. A substrate only is characterised by a single peak on an X-ray rocking curve. A single layered structure will generate two peaks, one for the substrate peak and a smaller layer peak. A crystal lattice with a change in composition between the bottom and top of a layer is a graded structure and will create a asymmetric peak for that layer. More complex crystal lattices include the MQW structure and super lattice, both of which have repeating twined layers, ABABAB and so on, grown on a substrate. The difference between the two types is that the MQW has a narrower A+B thickness. These complex structures produce the following features:

- . a substrate peak, usually the largest peak in the rocking curve.
- . a peak caused by the addition of Bragg reflections from the AB layers of MQWs (a zero-order peak).
- . a set of subsidiary 'satellite' peaks symmetrically surrounding the zero-peak, spacing determined by the periodicity (total thickness of the repeating layers) of the MQW, or interference from the gap fringes arising from interference between each of the layers comprising the MQW.

(D.K. Bowen 1989 in Interview with T.Tjahjadi)

In general, the rocking curve will have a peak for the substrate, one peak for each layer in the structure, and a combined peak with satellite peaks for any repeating layers present.

Structural Effects

Although a peak can be formed by an individual layer and will, in principle, give rise to a peak, peaks can also arise from interference effects between layers. Interference can be either positive or negative. Positive interference can result in additional peaks, whilst negative can cancel out a peak. This makes reading a curve more difficult than might at first be expected. Table 5.1 outlines some additional predicted effects of structural parameters on the shape of the rocking curve profiles.

These structural effects are taken into account in the simulated model, and matching the experimental rocking curve

to the model begins by entering the expected experimental structure into the simulated model, producing a model, and then comparing the simulated output to the experimental rocking curve. Any differences between the two will be the result of the experimental rocking curve deviating from the expected structure. X-ray rocking curve analysis involves changing the structural parameters of the simulated model until a model is produced that matches the experimental structure.

Table 5.1
Structural Parameters and their Effect on the
Rocking Curve Profile.

(D.K. Bowen)

Structural Parameter	Effect on Rocking Curve
Layer thickness	Peak height
Grading in a layer	Peak asymmetry
Composition of layers	Peak position, and width
Very thick layer	Peak width
Very thin layer	Moves peak close to substrate peak
Repeating layers (AB)	A Zero order peak with satellites
Periodicity (A+B)	Number/spacing of satellite peaks

Epilayer Defects

There are a number of epilayer characteristics that have a direct effect on the corresponding layer peak in the rocking curve. Generally speaking, it can be said that when the epilayer is defective if it broadens the layer peak. A misorientated epilayer will split a peak and the same can be said of a mismatched epilayer. Non uniformity in the epilayer both broadens and splits the peak, whilst curvature in the sample will broaden the substrate peak and all associated layer peaks. Tilt of the layer with respect to the substrate will shift the expected position of the layer peak.

All of these defects in the epilayer are the result of the growth process, and this can not be predicted. If these uncertainties are added to the interactive effects that can occur between peaks it is apparent that a complex rocking curve profile does not directly correspond to the expected originating structure. Expertise is required, therefore, to isolate particular interactive effects such as interference fringes, satellite peaks and hidden peaks in order to match the experimental rocking curve to the simulated model.

To a certain extent defects in the epilayer can be predicted in accordance with the composition of the crystal lattice. For example, when a layer with a radically different lattice parameter is grown on a substrate then a biaxial strain is created which can introduce curvature into the sample. This is particularly evident when the layer is very thick and in these circumstances it is not possible for coherence to be attained, so the layer tends to relax, creating a mismatch

between either the supporting layer or substrate. In situations where the mismatch is severe, and layers do not relax tetragonal distortion results. The degree of mismatch can be calculated in the following way where the unrelaxed mismatch is:

$$M^* = \frac{\text{Change in lattice spacing}}{\text{Divided by lattice spacing}}$$

$$= \text{Peak splitting } X \cot (\text{Bragg angle})$$

Whilst the relaxed mismatch can be found by:

$$M = M^* \times \frac{1 - v}{1 + v}$$

where v is the poisson ration and x is given by:

$$x = \frac{m}{M}$$

where m is the measured relaxed mismatch and M is the mismatch between two binary components of a tertiary alloy. The effects of misorientation can be predicted by simply rotating the specimen 180° from its optimum setting and assuming misorientation to be exactly twice the angle between the reflecting plane and the specimen surface. Both of these factors can be modelled and incorporated in the simulated rocking curve. Unfortunately, tilt has not yet

been modelled in the simulation software, and the effects of epilayer roughness, or other such defects cannot be calculated directly, and only when all other factors are isolated can the simulated and experimental curves be visually compared to detect such effects.

5.2 Is X-ray Rocking Curve Analysis a suitable domain for modelling within an expert system architecture?

It is only necessary to develop an expert system if the task(s) to be solved are of a certain type. According to Forsyth there are a set of criteria that can be applied to the domain to decide if it is suitable for expert system development (Forsyth 1984). These are given in Table 5.2.

Table 5.2
Suitability of Domain Characteristics for Knowledge-based
Approach

Suitable	Unsuitable
Diagnostic	Calculative
No established theory	Well defined formulae
Human expertise scarce	Expertise widespread
Data very noisy	Facts known precisely

In applying the diagnostic criteria to X-ray rocking curve analysis, it is clear that the domain can be considered a problem solving task. The double X-ray diffractometer produces a spectrum reflecting the structure of a sample which can be investigated for faults using a simulated model. Most medical expert systems such as MYCIN and INTERNIST (see Section 2.4) are also diagnostic in nature producing models of the disease, and have also been used in image processing for feature extraction (Matsuyama 1984), two diverse yet applicable areas to X-ray Rocking Curve analysis. There are established diffraction theories that can be used to produce models of a crystal lattice (see Section 5.1.1). However, the growth process of a crystal lattice is not fully understood, and defects in the crystal structure are common (see Section 5.1.2). Most faults can be modelled mathematically, and simulation used as a matching criterion for the experimental rocking curve. The problem is knowing which faults to model. In this respect, there is no established theory for choosing these and no possibility of developing one. The second criterion, therefore, applies. The third criterion is the availability of expertise. Included here is the both complexity and criticality of the diagnostic task. There is a shortage of experts for the domain, less than one per diffractometry laboratory. Furthermore, training to become an expert takes a long time, and degree level theory required to understand the process is substantial. The turn around of samples in the laboratories can be as many as a thousand samples a day, and there is a marked difference in the iterative simulation

cycle times between experts and novices. The final criterion is the degree of uncertainty attached to the knowledge of the domain. In this respect, there are precise facts such as the effects of the structural factors on the rocking curve profile, or certain epitaxial defects such as curvature. These have been modelled to produce a precise rocking curve definition. However, the exact structure and their combined effects on the rocking curve profile are never known. Rough guesses have to be used by the experts when iterating down to a match.

In all respects, the chosen domain is suitable for expert system development, being very similar in nature to many other diagnostic tasks solved using knowledge based techniques.

Part Two (Analysis of the Domain)

5.3 A Cognitive Analysis of the Domain

In formulating an overall design for the expert system there are many A.I. techniques that could be used in structuring and reasoning with the domain. It was decided to assess the overall characteristics of the domain and attempt to fit existing A.I. methods to the problem, rather than create a specialised technique specific to the area of X-ray rocking curve analysis. Four important characteristics of the domain were identified, each of which is common to most expert system designs (Davis and Lenat 1982). They included the identification of the inputs and outputs of the domain, the

utilisation of search systems that could be applied to a problem in the domain in unstructured situations, inference to resolve problems and representation to state the problem.

5.3.1 Input and Output.

This is the type of data that will be available to the system before and during a consultation, and the type of solution required by the user. Of particular relevance here is whether the inputs are numeric or non-numeric or mixed, certain or uncertain, volunteered or user interactive or both, and whether the output required is the best solution or series of close approximations, and critical or non-critical to the evaluation of the problem. It is also necessary to know the level of knowledge the user has in operating the system, and, therefore, the degree and type of explanation required by the program interface.

5.3.2 Search

At the general level it is useful to conceptualise the problem domain as a problem space described in terms of its depth and breadth; thus, one talks about the complexity of the domain as a measure of the depth of a problem from a start state to a goal state (number of changes in state to solution), and the size of the domain as its breadth at each level of the solution (number possible alternative changes in state), both of which vary according to the type of problems encountered (see Section 3.2). In general, the more complex a problem the greater the need to structure the knowledge of the domain. Domain problems can be further

divided into separate stages or intermediate states between the start state and goal state, and here one is concerned with the interdependency of the stages in processing towards a solution, and whether the steps to solution can be ignored, undone, or remain fixed (see Section 3.3). The general suggestion of the A.I. community is that the control strategy used to search through the problem space needs to be progressively more complex as the intermediate steps to solution become increasingly irrecoverable using respectively simple recursive programming, push-down stacks or complex planning procedures (Rich 1983). Heuristics can be brought into play as a means of reducing the size of the search space and a range of techniques relevant to the domain can be used to map onto a problem state a degree of acceptability, possibly in the form of a numerical analysis. A range of heuristics are available for this purpose including generate-and-test, means ends analysis, hillclimbing, and possibly specialised search techniques such as genetic algorithms (Lenant 1983).

5.3.3 Inference

When reasoning about a problem, it is important to know if the system will be dealing with a finite or infinite set of possible solutions, and whether or not novel situations will be encountered. In other words, whether the selection of hypotheses for solving a problem will be dependant on logical or statistical pre-requisites. Here there are a number of principles that become relevant to expert system design including: resolution, bayesian logic, probability

theory. Such systems can be incorporated into the control structure of an expert system and used determine how the knowledge of the domain is used.

5.3.4 Representation

In applying knowledge to a problem the needs of search are reduced. Widely recognised techniques for knowledge representation include predicate logic, semantic networks, frames, and rules. There are also a a range of methods available for applying that knowledge to a specific problem including logical inference resolution, production rule systems and statistical inference. The degree of structuring of knowledge has to be established when using such techniques and decisions made as to when knowledge systems might be used as opposed to search systems in solving a problem. It, therefore, follows that the way the knowledge of the system is structured and used by the expert is crucial to the selection of representation techniques, and this directly influences the way search systems are activated when knowledgeable techniques fail to resolve a problem. The types of objects that will be described in the knowledge structure have to be specified along with the inter-relations to other objects in that domain, and this again is reflected in the type of knowledge representation chosen.

Selection of Techniques

5.3.5 Input and Output Characteristics

How data are input to the system and what outputs are required, influence the choice of A.I. techniques. Generally, input will include numeric values that correspond to known equations in the domain which can be used to characterise a curve. For example, if the peak splitting is more than three times the width of the larger peak then it is possible to deduce the structural parameters of the rocking curve without the use of simulation. Likewise, the relative areas under layer and substrate peaks can be used to calculate the thickness of the layer and so on. Knowledge of this type provides powerful inputs to the use of algorithms that in turn provide input to a constraint propagation system. Demon logic is one formalisation of such a system with the actions being dependant on the satisfaction of all pre-conditions which can be sustained at any time during the process. This "watching" characteristic gives the reasoning a non directional character, which can be useful if a consultation becomes fixed on one solution for too long.

The user of the system is not always likely to be knowledgeable, and, therefore, the system may have to reason with uncertain data. There are three solutions to this situation:

- a) Add consistency checks to data, increasing redundancy.
- b) Allow certainty factors to be used when reasoning.

c) Tailor the knowledge to the level of the user.

The modelling of uncertainty is an essential element of the reasoning strategy and the use of probabilistic knowledge may provide a way of handling certainty, and the approach selected for this project. Lindley argues that through the development of axiomatic systems and scoring functions, probability should be able to handle uncertainty without the need for fuzzy logic, belief functions and so on (Lindley 1987). More specifically, Nilsson puts forward ideas about the combined use of logic and probability theory which may prove useful when considering alternative inference mechanisms (Nilsson 1986). The tailoring of knowledge by assessing the responses to questions posed by the system to the user will also be investigated, but as a separate issue. The use of built-in redundancy will not be explored as an option.

There are two types of input to the system, an initial volunteering of known data about a crystal, followed by an interactive session with the user. This indicates that a mixed search strategy (sideways chaining) should be used when pursuing each separate sub-goal; means-ends analysis is one such option (Enst and Newell 1969). Another important aspect of system inputs and outputs is that the reasoning strategy of the system should follow the logic that would be used by the expert. This is necessary because the system should have a training element within it, where the user has feedback from the system which allows understanding of the X-ray crystallography analysis to develop. The causal

modelling of the system is important and will be emphasized in the design of the inference engine. Production rule methods tend not to provide such a capacity, whilst logic based programming does (Kahn 1984). However, in questioning a user about the characteristics of a simulated curve, it is important that when the system gathers evidence, the ordering of the consultation appears natural. The experience of the MYCIN project was that by tying the questioning strategy closely to the reasoning process the gathering of information appeared scattered. The reasoning of an expert does not necessarily result in the generation of reasoned questions, but rather topic orientated questions. A separate questioning strategy is recommended for this expert system (Buchanan and Shortliffe 1984). The user may wish to know why certain lines of reasoning were followed and what certain choices mean. Causal modelling provides "logical" reasons, and is favoured over more shallow techniques as it can provide justifications for the choice of certain options (Kahn 1984). Canned descriptions could be included as a option for further information during a session without much expense.

5.3.6 Inference

When reasoning within the X-ray crystallography domain it is apparent that it is an open-ended diagnostic task where one cannot assume the uniqueness of a theorem. In trying to match an infinite number of simulated curves to one experimental curve it is probable that a novel interpretation of the data will arise that cannot be

predicted at the start of the diagnostic process. This indicates that a causal reasoning process is favoured over a shallow reasoning, and backed-up by the assertion that the domain is underpinned by well understood principles of physics. The probability that the system will be required to analyse novel structures is further evidence in favour of the use of "deep reasoning", and tends to detract from the use of production rules since they tend to favour closed systems of knowledge where all possible outcomes are encoded in the rule base. Reasoning with logic, uses deductive methods to assert new information and can function beyond data. Support for both these views can be found in the work of Kahn (1984) and Davis (1980). Further support for the use of a causal model can be derived from the nature of the problem solving process. In the steps to solution, it is possible to uniquely identify specific characteristics that make-up the overall structure of an X-ray spectrum. In the case of a double diffraction profile from a MQW structure these include a substrate peak, a MQW peak, layer peak(s), and satellite peak(s). Each of the parameters for each layer can be altered in turn to optimise the matching process between the simulated curve and the experimental curve, thereby, systematically arriving at the best solution. The decompositional nature of this reasoning process again suggests the use of causal modelling, and since it is not possible to know at the outset all the possible solutions to the problem, prenumerated solutions must be discounted in favour of a constructed solution technique using methods such as constraint propagation that limit the selection of

hypotheses by the placing restrictions on the viability of certain paths of reasoning (Buchanan 1982).

The expectation as to the initial composition of the crystal is known from the outset, although the outcome of the growth process is uncertain. Certainty is achieved through what might best be described as an 'iterative' knowledge guided procedure where trial structures are generated again and again until a best fit is found. This makes the formation of a strong initial hypothesis an important factor in the experts reasoning, with frequent feedback from the outcome of the process determining the final solution. In terms of the maintenance of hypothesis a best hypothesis is a better method handling competing solutions rather than the maintenance of a series of candidate hypotheses (Lenant 1983).

5.3.7 Search

Search strategy is an important aspect of reasoning when knowledge begins to run out, and if one considered the complexities that can arise from interactive effects between the refracted X-rays from layers in a crystal, the data will be noisy with guidance to the solution hindered by successively ambiguous interpretations. In such situations it is better to carry data forward in a process rather than deduce backward from a solution. Further to this, Baucanan (1978) suggests that opportunistic search techniques are used under these circumstances, working outwards from an "island of certainty", which in the case of the X-ray output would be each known peak produced from the crystal.

However, in reasoning about a rocking curve profile, it is unlikely that strong pieces of evidence will become available during inferencing that suddenly requires the system to pursue another hypothesis, which tends to mitigate against both complex control and the use of forward chaining. Under these circumstances backward reasoning would be favoured.

It has been suggested that the limited scope of the domain and the decompositional nature of the reasoning allows tasks and sub-tasks to be easily defined. Yet the order in which they are pursued can be critical to the matching of the experimental curve with the simulated curve. The selection of tasks is often dependant on the knowledge of the expert and cannot be left to the reasoning process alone. This implies that simple control rules such as forward or backward chaining or sideways chaining could be used to reason within sub-tasks, but not between sub-tasks. The use of a scheduling system or blackboard could be maintained for this task as for speech signals (see Section 2.3.3), but this demands high resources and for the limited analysis of an X-ray spectrum as compared to a speech wave would seem unnecessary. The use of an agenda is less expensive to maintain, and sub-tasks could be re-ordered on the basis of control rules that express critical knowledge of the domain.

5.3.8 Knowledge Representation

The final considerations are the methods of encoding expertise in the knowledge base. There are four types of

representations available and of these large production rule systems can be omitted since the domain of understanding is both highly structured and interlinked. However, there are small parts of the domain that suit this type of representation. There is a predictable hierarchical relationship between the elements in a X-ray spectrum, and this suggests that either semantic networks or the more specialised frame system could be used to represent the knowledge. The inheritance of properties is an important aspect of the system and, thereby, further supports the selection of an object orientated representation. However, frames, in the form of a slot/filler notation, are preferred over semantic nets as they can be easily integrated into multi-level systems. In this instance, the filling of a slot may require the system to seek a calculated value, or assign a default value in the absence of data or call a super-ordinate or sub-ordinate frame for data outside the scope of the concept (see Section 4.2.3). Logic representations may be used to represent knowledge, typically as a binary predicate. However, whilst it is possible to overcome the problems associated with the monotonic reasoning by using reversible logic such as Default reasoning, Modal logic and Autoepistemic logic, the limited expressive properties of this type of representation remains. Frames and semantic networks express knowledge in both a declarative and procedural way and are, therefore, a better way of linking symbolic code to the cognitive representations. This is backed up by the fact that cognitive structures tend to be object centred (Klasky 1975). Finally, through analysing the

problem solving procedure of the expert, it is apparent that when re-running a simulation against an experimental rocking curve, the differences between each simulated output is an important criteria used when adjusting the lattice parameters on the next simulation. This suggests that it is important to express concurrently both the common features of objects, but recognise their differences. Buchanan and Shortcliff state that under these circumstances objects are best represented using a frame system (Buchanan and Shortcliffe 1984). Table 5.3 gives a summary of the A.I. techniques selected for the project outline above.

Table 5.3
Summary of Techniques used within Expert System

Problem Structure	A.I. Method
Question Strategy	Topic Orientated
Overall Control Strategy	Causal Modelling
Control Interface	An Agenda
Direction of Reasoning	Mixed (forward/backward)
Knowledge Representation	Frames and Production Rules
Hypothesis Maintenance	Best Hypothesis
Type of Reasoning	Logical Inference and Demons
Uncertainty	Probabalistic Logic

5.4 The Expert System Core

The expert system for the analysis of X-ray rocking curves is composed of six elements: a user interface for arranging the questioning strategy, a knowledge base consisting of a frame structure describing the rocking curves and the structural parameters and a production rule system for backward chaining from a procedural attachment in the frame system, external procedures for mathematical calculations called by the frame system, a database for storing facts and the results of the instantiation of the frame structure, an inference engine that utilises inference to matches the goals of the system to the knowledge in the frames, producing a best hypothesis working from the principles of probability, and a control system based on causal modelling that orders the completion of tasks on the basis of a priority system organised as an agenda. The overall structure of the expert system is shown in Figure 5.5.

5.4.1 Knowledge Base

A mixed representation has been implemented for the X-ray rocking domain consisting of a frame hierarchy to represent a rocking curve taxonomy, and a set of production rules representing general experimental conditions. Constraints have been added to form part of the procedural knowledge of the frame system. Default knowledge is also available within the frame hierarchy.

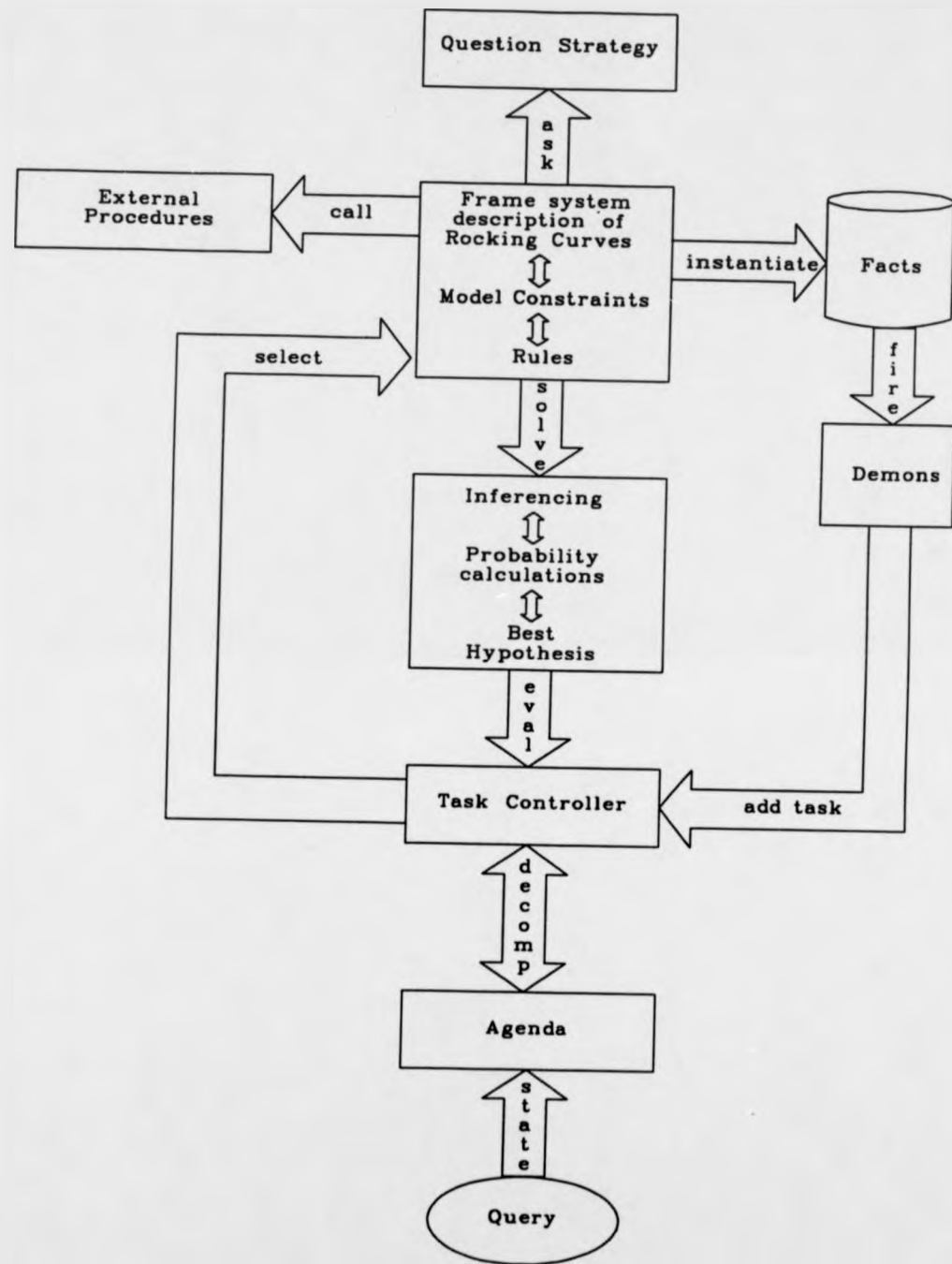


Figure 5.5 Expert System Architecture for X-ray Rocking Curve Analysis.

5.4.1.1 Frames

The frames represent the objects of the domain. They are a data structure that allows a typical instance to be generated once knowledge is added to the frame. The frame is schematic and represents objects within the domain. In the current system, each frame in a domain has a common structure based on the filler slot notation (Thayse 1988). The frames are represented in LISP as a nested associated list structure. This means that a frame is ASSOCIATED with a series of slots, each of which is ASSOCIATED with up to three types of facet, each of which has one or more values attached (see Figure 5.6).

```
[ <frame name>
    ( <slot 1>
        ( <facet 1> )
        ..
        ..
        ( <facet n> ) )
    ..
    ..
    ( <slot n>
        ( <facet 1> )
        ..
        ..
        ( <facet n> ) ) ]
```

[] = Frame definition

() = Nesting structure

<> = variables

Figure 5.6 Nested List Structure of each Frame

The slots relate to possible descriptions of a schema or frame and the facets the manner in which that description is

achieved. In the system three types of facet are used, VALUES that are the declarative descriptions of the slot, DEFAULTS which are the most likely declarative descriptions of the slot, and IF-NEEDED procedures which are procedural attachments that acquire the description through procedural knowledge. The reference crystal frame is shown in Figure 5.7. It shows a frame before instantiation. There are four data slots and one control slot in the structure. The data slots are filled in the order specified in the control slot. A SYMMETRIC geometry with a 004 reflection indices are both default values for data slots 2 and 3, with constraints placed on any entry outside these assumed values.

```
(REFERENCE-CRYSTAL (frame)
  (AGENDA (control slot)
    (COMPOSITION REFLECTION-INDICIES GEOMETRY
      EFFECT (names of active slots)))

  (COMPOSITION (data slot 1)
    (IF-NEEDED {facet} FASK {procedure}))
  (GEOMETRY (data slot 2)
    (DEFAULT {facet} SYMMETRIC {value})
    (IF-NEEDED {facet} FCASK {procedure}))
  (REFLECTION-INDICIES (data slot 3)
    (DEFAULT {facet} 004 {value})
    (IF-NEEDED {facet} FCASK {procedure}))
  (EFFECT (data slot 4)
    (IF-NEEDED {facet} RULE {procedure})))

() = nesting structure
{} = comments on structure
```

Figure 5.7 A Typical Frame Structure for Expert System.

The EFFECT the reference crystal has on the shape of the rocking curve is determined by the production rule system. The COMPOSITION of the crystal is acquired either through the inheritance of a value, otherwise through an unrestrained input from the user.

Knowledge in the frame system is divided into two types, declarative knowledge and procedural knowledge. Declarative knowledge is stored as a list of values or possible default values inherited, as necessary, from within the frame system. The procedure knowledge exists as a series of rules, and constraints that are applied if no declarative knowledge is available. During the consultation the frame hierarchy, which initially has no values, is instantiated and the schematic hierarchy converted to a representation of the current expert system problem. The representation can be saved at any point during the running of a consultation, and because the property list(s) points to the originating schematic, the values found for the consultation can replace the schematic representation with the declaratives already found.

5.4.1.2 Production Rule Knowledge

Non-hierarchical knowledge is stored in a production rule format (see Section 4.2.4). Each rule is stored as a flat list within a RULE frame as a list of unique slots (see Figure 5.8).

Frame name: (RULES-RC

```
(RULE1
  (VALUE IF LAB-SET-UP-STRUCTURE HAS (NOT CUBIC)
    THEN
      MILLER-INDICES IS 001))
(RULE2
  (VALUE IF LAB-SET-UP-WAVELENGTH HAS SYNCHROTRON
    THEN
      REFERENCE-CRYSTAL IS UNNECESSARY))
(RULE5
  (VALUE IF LAB-SET-UP-LAYER-PARAMETER HAS SPLITTING
    AND LAB-SET-UP-LAYER-PARAMETER HAS (NOT COMPOSITION)
    AND LAB-SET-UP-LAYER-PARAMETER HAS (NOT RELAXATION)
    AND LAB-SET-UP-LAYER-PARAMETER HAS (NOT EVEN-THICKNESS)
    THEN
      REFERENCE-CRYSTAL IS UNNECESSARY)))
```

Figure 5.8 Three Rules from the Storage Frame for the Production Rule Knowledge Base.

Each rule in the knowledge base occupies a value facet that is associated with a unique slot label composed of the prefix RULE followed by a positive integer. All production rules for the specified domain occupy the same frame and are consequently stored together in the same knowledge base. Each rule is a list of atoms that conforms to a production rule structure. At the top level the rule is divided into a two part IF ... THEN ... structure called the antecedent and the consequent. It states that IF the antecedent is true THEN the consequent will follow. This representation is used by the modus ponens (MPP) rule of logic that says that IF x (and x then y) THEN y. The MPP rule is used in the inference engine of the production rule system using propositional logic (see Section 4.3.1). The antecedent can be further split into individual clauses that are connected together by

binary operators. There three binary operator available (AND OR XOR) and each clause is paired to the previous one by the operator such that (W xor X or Y and Z) reads (((W xor X) or Y) and Z). The validity of the complete proposition depends on the rules of propositional logic. The consequent has the same structure as the antecedent, but only the AND binary operator is available to this part of the rule. Each individual proposition is also divided into a tuple composed of an Identifier, Relation, and Value. Figure 5.9 gives a breakdown of RULE5 from the production rule knowledge base.

Propositions		Binaries		MPP
IF				
L HAS S	AND	Clause 1	Antecedent	
L HAS (NOT C)	AND	Clause 2		
L HAS (NOT Rx)	AND	Clause 3		
L HAS (NOT E)		Clause 4		
THEN				
R IS U		Clause 1	Consequent	
Singular Propositions (or tuple)				
Identifier	Relation	Value		
L	HAS	S		
L	HAS	(NOT C)		
L	HAS	(NOT R)		
L	HAS	(NOT E)		
R	IS	U		
L=LAB-SET-UP-LAYER-PARAMETER R=REFERENCE-CRYSTAL				
S=SPLITTING C=COMPOSITION Rx=RELAXATION E=EVEN-THICKNESS				
U=UNNECESSARY				

Figure 5.9 The Structure of a Single Production Rule

Each rule is stored and accessed sequentially unless a specific key, ie the rule names (RULE5), is used during inference. Each rule can have as many binary operators as required, although it is more efficient to have many small rules rather than fewer larger rules.

The constraints frame consists of a series of slot entries corresponding to existing frames within the domain, and nested facets entrie(s) corresponding to slots in the frame (see Figure 5.10). When operating constraints, each time a FCASK procedure is encountered in the procedural knowledge of a slot the frame reasoner passes the calling frame value and the corresponding slot values as arguments to the FCASK routine that then binds them to the parameters FRAME and SLOT.

```
Frame name: (CONSTRAINTS
(LAB-SET-UP (WAVELENGTH SYNCHROTRON Mo Fe Ag Cr)
            (MILLER-INDICES 111 222)
            (ARC-RANGE 100 TO 40000))
(LATTICE   (SUBSTRATE GaSb Si Ge)
            (LAYERS AlAs InGaAs InP)))
```

Figure 5.10 Part of the Constraint Frame Structure

The procedure is then executed and searches the CONSTRAINT frame to see if a match can be found for these values. For example, if a FCASK procedure is activated in the frame LAB-SET-UP for the slot WAVELENGTH the FCASK procedure will find a match in the constraints of "SYNCHROTRON Mo Fe Ag Cr". The

procedure will then inspect the question frame using the same frame and slot values to see if a canned question exists for this call. If a question exists then it is returned and posed to the user, if not, one is generated from the frame and slot values. Once the question is posed the procedure then configures the constraints into either a series of options with option numbers or a range statement depending on the format of the constraint. The user response is then restricted to the returned constraints.

When the user replies to the constraints any options included in the reply are added to the slot of the calling frame as atomic entries. These values are also added to the IDENTIFIER frame of the shared database as a multiple HAS relation under the composite name of the frame and slot that generated the constraint. All constraint options not selected by the user are also added to the database under the same slot entry, but negated. All carriage returns are treated as NIL responses and will exit constraints on a NIL value. This means that no constraints will be added to the slot's value facet of the calling frame, and all the constraints will be returned as negative entry (NOT x) to the identifier frame.

Constraints are always applied sequentially in this five stage process and operate only when procedural knowledge is required via the FCASK attachment. All knowledge generated from the application of constraints is stored in the database as a multiple HAS relation. This is necessary because constraints can always have more than one value. The HAS type relation is also used to store the results of

instantiation in all other parts of the frame system, and, therefore, all database entries derived from the frame system are of a list construction.

5.4.2 Inference Engine

There are three inference engines in the expert system. There is a frame engine that chains forward, a production rule engine that chains backwards, and a demon logic engine that chains forward. The frame engine controls the consultation, and only suspends operation when either a RULE attachment is called, or a new fact is added to the database. The production rule system takes control when called via procedural attachments and demon logic is activated every time the database is updated.

5.4.2.1 Frame Inference Engine (The Agenda)

The agenda is the main controlling device that is used to govern the complete consultation. The agenda is made up of three sections and processing moves from the input section across to storage section via a set of calculations. The agenda receives inputs from the frame system in terms of slots for which values must be found, and assigns priority levels to each slot based on their position in the control slot of each frame. The slot assignment is determined by a frame called the DICTIONARY, which stores slots in order of priority. Control is then passed to the frame system which finds values for each slot and returns the results to the

agenda. Based on the ratio of success to failure when finding slot values, the priority level of the tasks and subsequently any slots assigned are adjusted.

AGENDA			
CALCULATIONS		ANALOGS	DATA
TASK A L L O C A T I O N	P R I O R I T Y A S S I G N	TARGET	Tasks complete
		SOURCE	Tasks incomplete
		MAP	
		EVALUATE	tasks to be completed

Figure 5.11 The Structure of the Agenda or Controller.

Because the agenda only passes the top priority slots back to the frame system for filling, slots with a lower priority will only be filled towards the end of the consultation. However, because the priority levels are always being re-assessed upwards if positive slot values are found for their associated slots, and downwards if NIL slot values are found for their associated slots the order in which slots are queried will change during a consultation. Figure 5.11 outlines the agenda in detail.

When a consultation begins the agenda is activated by putting an agenda frame into the environment as a property list. This means that if the knowledge is saved and the consultation abandoned it can be restored to its original status by loading the re-saved knowledge package.

No tasks are assigned to the agenda at this stage. To control the sequence in which tasks are found a data dictionary is used. Again, this has a program entry in the knowledge package and when the agenda is activated the data dictionary is put into the environment as a property list (see Figure 5.12).

Frame name: (DICTIONARY-RC

```
(LAB-SET-UP (VALUE WAVELENGTH STEPS SCAN)
              (VALUE PEAK-COUNT ASYMMETRY PEAK-HEIGHT))
(SUBSTRATE  (VALUE MATERIAL ORIENTATION)
              (VALUE HALF-WIDTH)
              (VALUE DESCRIPTION)
              (VALUE PEAK-SHAPE)))
```

Figure 5.12 The Structure of the Data Dictionary

The agenda and frames system are the critical components in the expert system that search the frame system frame by frame until the slots are satisfied. The agenda inspects the current frame for slots. When a list is found control is passed to the frame system which returns an ordered list of active data slots from the control slot of the frame. Control is then returned to the agenda. Each slot is then added to the to-be-completed part of the agenda and assigned a sequential decremental value so that the first slot is only decremented once, and second slot decremented twice and so on. The agenda then releases the slot with the highest priority to the frame system and passes control once again back to the frame reasoner. The frame system attempts to fill the slot using forward chaining by generating a search tree. If a value is found for the slot then it is placed in the Tasks Complete section of the agenda. If no value is found then it is placed in the Task Incomplete section of the agenda. Once the slot has been placed into one of these two sections it is removed from the to-be-completed section. The next slot from the to-be-completed section of the agenda is then selected by the control system and passed to the frame reasoner for filling, repeating the process. This control cycle of agenda -> frames -> agenda continues until no more slots exist in the to-be-complete section of the agenda.

The slot filling cycle is embedded in a outer frame satisfying cycle, which is dependant on the root generated from the top of the frame hierarchy to the query frame The list of frames created from the hierarchy are instantiated

in series, and the process is again controlled via the agenda. When there are no active data-slots belonging to the current frame agenda control is switched to the next frame in the frame-list. The outer cycle is again a data driven process, with the facility of hierarchical inheritance operating within the frame reasoner, hence propagating values further down the frame-list if necessary, before the agenda reaches that point. The two agenda cycles are summarised in Figure 5.13.

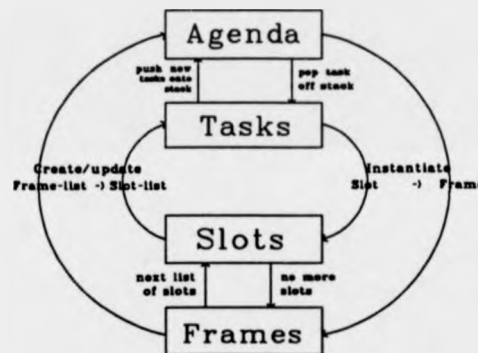


Figure 5.13 The Control Cycle for the Agenda and Frames

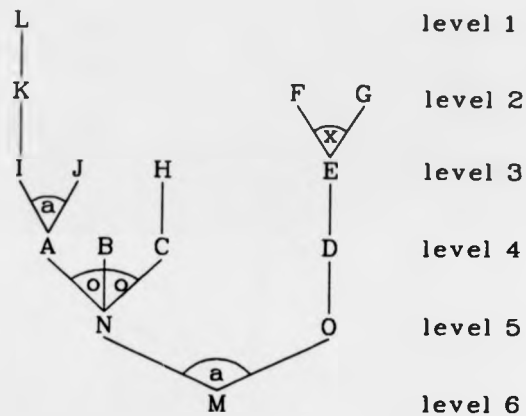
5.4.2.2 The Production Rule Engine

When procedural knowledge is required via the RULE function, the value of the calling slot is passed to the production rule system, and becomes an identifier requiring a value. The aim of the production rule engine is to find a rule that

contains an identifier with this name, and then prove that the selected rule is valid. If the rule is valid the value associated with the identifier is returned to the frame system and used to fill the slot. If the rule is not valid then another rule is sort with the same identifier name. The production rule system continues to search for rules that match the identifier name by searching sequentially the RULE frame until either no more rules are available or a valid one discovered. If none prove correct then NIL is returned by the production rules system, and no entry is made in the frame system of the calling slot for the RULE procedure.

The inferencing process is slightly more complex than indicated above because, facts are not always available in the database; and the propositions in the antecedent often backward chain to other rules. When a fact does not exist in the database, the inference engine will try to establish it by checking to see if there are any other rules in the system that might imply this fact, otherwise it will generate a question, asking the user if a particular proposition is true. This process continues until either all database entries are found for the end of a chain of antecedents or no more matches are made. Any propositions that are not supported by database entries at the end of the chain are formed into questions posed to the user. Once all the database entries are collected the complete premise is put to the prover, and is returned as either a valid or invalid formula according to the rules of propositional logic. The complete chain of rules is what is meant as the rule-tree and is shown in Figure 5.14. It shows a typical

rule-tree in which the goal or consequent M in the rule IF N AND O THEN M is linked to the rule IF A OR B OR C THEN N and IF D THEN O respectively which are themselves linked to the rule IF I AND J THEN A for A, B via a database entry, and C by the rule IF H THEN C. D is linked to the rule IF E THEN D and so on up the tree.



Where:

A ... O = Propositions
a = AND binary operator
o = OR binary operator
x = XOR binary operator

Figure 5.14 Example of a Backward Chaining Rule-tree

In total, the goal M is inferred by a set of data-base entries formed from propositions B, J, and G, and a set of user responses to formed from propositions L, H, and F, all of which are linked back to the goal via a chain of IF ... THEN ... implications stored in the knowledge base. The complete premise for this rule-tree is:

IMPLICATIONS:

[[((((([((((((N and O) implies M) and (((A or B) or C) implies N)) and (I and J) implies A)) and (K implies I) and (L implies K)) and (H implies C)) and (D implies O)) and (E implies D)) and ((F XOR G) implies E)] and ~

FACTS:

L) and J) and (NOT B)) and (NOT H)) and F) and (NOT G)] ~

GOAL:

implies M]

This premise returns a valid formula for the rule-tree since the complete truth table for the propositions, given the goal M, results is each implication being true. If the same implications and goal is used as above, but the facts changed to:

L) and J) and (NOT B)) and (NOT H)) and F) and G] ~

a NIL result would be returned since proposition G is now positive, and from the rule-tree in Figure 5.14, F XOR G would return a NIL result which would propagate down through single implications to O, and because proposition M is dependant on the binary antecedent O AND N, goal M would be returned as invalid.

The database of the expert system is modified as a result of each inductive process, and in the case of all rule-trees the root goal and those end-branches that generate questions are changed. All end-branches that do not already have database values have the proposition added to the system in tuple form. If the goal is achieved then the proposition is also added to the database in tuple form. In the case of the rule-tree in Figure 5.14 and the induction immediately succeeding it, the database has the propositions L, H, F, and M added to the database. If these propositions represent the tuples L IS 1, H IS 2, F HAS 4, and M IS READY, then identifiers are set as follows:

```
L SETQ 1
H SETQ (NOT 2)
F SETQ (4)
M SETQ READY
```

Goal M is set to the value of the proposition so that the if proposition M is included in any further rule-trees, it will not be necessary to re-generate it. It is also a characteristic of the production rule system that even when

a premise proves invalid the propositions at the end of the branches that generated questions, ie L, H, and F, are still added to the database. This means that even a failed goal may have an effect on future inferencing.

5.4.2.3 Demon Logic

Demon logic is used as an interrupt mechanism, to change the reasoning or the control direction of the system given certain conditions. The principle was first introduced as a control mechanism in complex problem solving such as pattern recognition by Lindsay and Norman (1972) in a model called Pandemonium. Recently, it has been used for most notably in the HEARSAY-II project and used in conjunction with a blackboard style of control (Erman, Hayes-Roth, Lesser and Reddy). In this expert system design, demons have been programmed to operate as a forward chaining interrupt. Each demon is composed of an antecedent and consequent configured as a flat list of the form WHEN ... THEN ..., and all lists are stored in the DEMON frame.

Frame Name: (DEMONS-RC

```
(DEMON1
  (VALUE WHEN REF-CRYSTAL IS (NOT SECOND-CRYSTAL)
    AND STRUCTURE IS APPROXIMATELY-RIGHT
    AND SIMULATION IS RECOMMENDED
  THEN REF-CRYSTAL IS SECOND-CRYSTAL
  AND (PREP 'REF-CRYSTAL 'INCLUDE 'SECOND-CRYSTAL)))

(DEMON2
  (VALUE WHEN PEAK-HEIGHT IS APPROXIMATELY-RIGHT
    AND INTERFERENCE-PROFILE IS APPROXIMATELY-RIGHT
    AND PEAK-POSITION IS APPROXIMATELY-RIGHT
  THEN STRUCTURE IS APPROXIMATELY-RIGHT)))
```

Figure 5.15 The Storage Frame for Demon Logic Rules

The configuration of these demon rules is the same as for the production rules, and an example is shown in Figure 5.15. This figure shows two demon rules that perform two separate actions if the rule is proved correct. DEMON1 sets the identifier REF-CRYSTAL to the value SECOND-CRYSTAL and then transfers this value to the slot INCLUDE in the frame REF-CRYSTAL using a procedure called FREP. In DEMON2, when PEAK-POSITION, INTERFERENCE-PROFILE and PEAK-HEIGHT are all APPROXIMATELY-RIGHT then the STRUCTURE is APPROXIMATELY-RIGHT. This later proposition is added to the database if all three conditional clauses are correct. Figure 5.16 shows the structure of each rule.

	Propositions	Binaries	MPP
WHEN	C IS (NOT S) T IS A I IS R	AND AND	Clause 1 Clause 2 Clause 3
			Condition
THEN	C IS S (EXT)	AND	Clause 1 Clause 2
			Action

Where:

C = REF-CRYSTAL
T = STRUCTURE
S = SECOND-CRYSTAL

Identifier

I = SIMULATION
A = APPROXIMATELY-RIGHT
R = RECOMMENDED

Value

EXT = (FREP 'REF-CRYSTAL 'INCLUDE 'SECOND-CRYSTAL)

Figure 5.16 An Example Structure of a Single Demon Rule.

The conditional part of any demon rule can be a single proposition or a set of propositions joined by binary operators. In the example given in Figure 5.16, there are three propositions joined by two binary operators. The type of operators available to the conditionals are AND, OR and XOR, and these operate in the same way as for production rules. All propositions in the conditional must be tuples. The action part of the rule can also be a single or multiple set of actions. Unlike conditional clauses, action clauses can only be joined by the AND binary operator. No other binary operator is acceptable. The actions differ from the conditional in that any LISP code can be used as a clause, and, therefore, any type of action taken when the rule is fired. This makes demon logic a control mechanism within the expert system.

As stated before demon logic is activated when any change is made to the common database. This means that if a change is the result of a demon action, the demons will re-activate themselves in a recursive cycle. This cycle will eventually cease because as each demon is fired it is excluded from further reasoning. However, the cycle will otherwise continue in a forward chaining manner unless either the next demon in the chain fails to fire, or the actions of a demon in the chain do not add a new identifier to the common database.

There is an important difference between the operation of demons and production rules. Demons do not gather information from the user by generating questions if it is absent from the database. This means that a forward chain of

demons will stop as soon as the demon checker finds that values do not exist for all the identifiers in the antecedent of the selected demon. Under these circumstances, control is returned back to the reasoning system operating at the time of the demon activation. A rule-tree for demons is shown in Figure 5.17, and illustrates the difference from the production rule-tree.

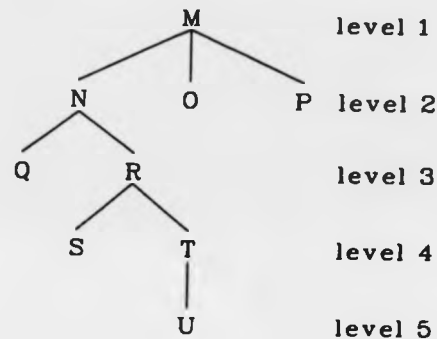


Figure 5.17 Example Rule-Tree generated by Demon Logic

In Figure 5.17 there are five calls to the demon system, represented by the five levels to the point where control is returned to the prior operation. All letters M-U represent action clauses from each of five demons. At level 1, M is a proposition added to the database. This generates three possible demon candidates (N, O, P), of which the first to pass the demon checker is the rule containing proposition N. This generates two candidates, the first failing, but the second, R, succeeding. At level 5 only one demon is matched

to the previous database proposition, resulting from the action at level 4, and this fails the rule check thereby ending demon control.

5.5 The Prover

The prover used in both the demon logic and production rule system operates by propositional logic. All rules in the knowledge base are stored as flat lists divided into antecedents and consequences. When ever a rule is formatted into a premise it is composed of implication(s), fact(s), and a goal. The task of the prover is to check that the premise is a well formed formula that conforms to the rules of logic. The prover uses the algebraic approach to logic using the rules of idempotence, commutativity, associativity, elimination, equivalence, involution, distributivity and De Morgans laws to produce the conjunctive normal form (See Section 4). The aim of the prover is to simplify the premise down to its conjunctive normal form. This is performed by firstly, dropping the AND connectives between the implications, facts and goal, so that they are seen as a separate formula. Then making the premise into an equation by moving the goal to the right and negating it. Then it is simply a question of trying each of the rules of logic to the formulae, moving, breaking down and expanding them until an axiomatic statement is left. Five examples of such rules are as follows:

a) $(A \text{ XOR } B)$ expands to $((\neg A \wedge B) \vee (A \wedge \neg B))$

b) $(A \rightarrow B)$ is equivalent to $(\neg A \vee B)$

c) $(A \wedge B) \Rightarrow C$ conversion to formulae $A, B \Rightarrow C$

d) $C \Rightarrow (A \vee B)$ conversion to formulae $C \Rightarrow A, B$

e) $\neg A, B \Rightarrow C$ dropping negation $B \Rightarrow C, A$

Where:

\Rightarrow = equal to

\rightarrow = implies

\vee = or

\neg = not

\wedge = and

$,$ = separation of individual formulae

The sequence in which the rules are applied is algorithmic, and follows standard logic procedures used in induction (Chang and Lee 1971), and follows the Boolean logic of the truth table. The system can handle any complexity of input, and is able to handle negative as well as positive instances of formulae. The system can, thereby, prove that something is not something.

5.6 Reasoning Methodology

The expert system combines production rules and frames to enable inductions to be made between slots remote in the hierarchy through the use of a production rule system that shares a common database, and by the selective accessing of

the production rule system through the controlling influence of slots. Generally speaking, the production rule system will only generate small rule-trees and return the proven values to the frame system without extensive search. The frame system will resort to the use of production rules when unstructured pieces of knowledge are required that can only be represented as IF ... THEN ... rules. The two systems also use different reasoning strategies. The frames operate using a data driven or forward chaining method, whilst the rules operate via backward chaining. The combining of both of these reasoning methods is recommended by A.I. researchers in the field because it counteracts the limitations of backward chaining, in that backward chaining tends to pursue a goal continuously without reference to data outside the immediate line of reasoning, and reduces the limitations of forward chaining, in that this method tends to be non-directional with no clear goal specified.

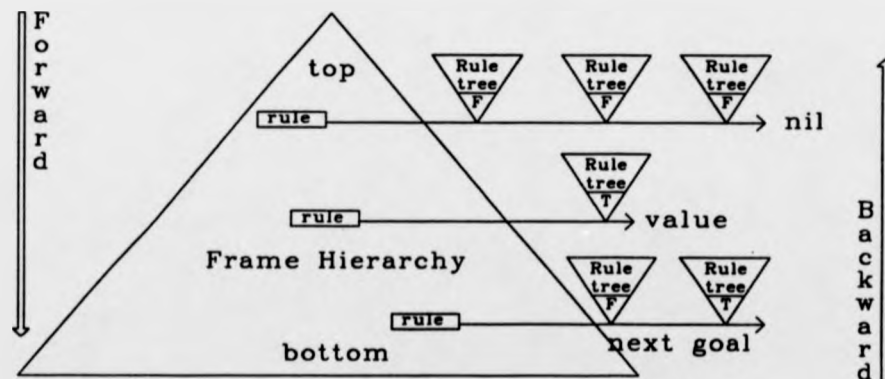


Figure 5.18 Sideways Chaining Method used for Expert System

The combining of backward and forward reasoning is sometimes called sideways chaining. A schematic diagram of the reasoning methodology is shown in Figure 5.18.

The sideways chaining overcomes the basic limitation of using one reasoning direction. However, demons have also been introduced as an interrupt mechanism to stop and then re-direct reasoning based on critical information. This is important in some applications since it is never certain when in a reasoning cycle critical information is either added by the user or inferred from the knowledge.

5.7 Database

All database entries are transparent to the user, but critical to the inferencing methods. They are made as a result of the following:

- a) Inheriting values for slots in the frame system
- b) Skipping or accepting defaults in the frame system
- c) Returning values as a result of procedural attachments
- d) Firing Demon rules
- e) Proving a production rule goal
- f) A valid user response to any generated questions

All database entries are recorded in two ways. Firstly, they are stored in a frame called the IDENTIFIER in tuple form: (IDENTIFIER RELATION VALUE). Within the tuples the identifier is a variable that is bound to a value. The relation determines whether the binding is singular or multiple. There are two relations used in the database IS

and HAS. The IS relation always holds a single atomic value unless it is a negative symbol in which case it is a list with a NOT-atom followed by the value. The HAS relation holds multiple values as a list which can be a mixture of negative symbols and symbols. For instance, if WAVELENGTH is an identifier bound by the IS relation then it might be any of the following values: NIL, (NOT Cu), or Cu. If LAYERS is an identifier bound by the HAS relation then it might be: NIL, (Si), ((NOT Si)), or (Si (NOT Ge) InP). The second way database entries are recorded is as free identifiers, with the values of the tuple bound to the identifier according to one of the two relational rules as outlined for the identifier frame. There is a fixed exchange of data entries between these two systems so that if an entry is made in the IDENTIFIER frame it is then transferred as a free identifier, and if a free identifier is created it is automatically added to the IDENTIFIER frame. For reasons of system design convenience, entries from the frame reasoner are made to IDENTIFIER frame, and free identifiers are created by the production rule system and demon logic.

The identifier frame exists as a program entry in the knowledge package and when the expert system is activated an empty frame called IDENTIFIER is created and stored as a property list. When the first database entry is added to the empty frame a pointer is created from the property list to the program entry, thus as identifiers are added to the frame the program entry acquires the same list structures. Figure 5.19 gives the frame structure for the IDENTIFIER frame.

```
Frame name:      (IDENTIFIERS
(WAVELENGTH      (IS Cu))
(LAYERS           (HAS Si (NOT Ge) InP))
(STRUCTURE        (IS (NOT CUBIC)))
```

Figure 5.19 Structure of Database Frame.

The integrity of the database is maintained because a frame entry cannot be added to the common database without it also being added as a free variable. Likewise, a free positive variable, a variable that has not been negated, cannot be created without also being added to the calling frame. If for any reason the system is interrupted and the consultation not resumed, a database integrity checker runs to ensure that both the frames and free variables are equivalent.

The structure of the database frame consists of a frame name called IDENTIFIER, cumulative slot entries each representing an identifier, an associated facet representing the relation of the identifier and a list associated with the facet representing the values of the identifier. Figure 5.20 summarizes the flow of database entries in the system.

5.8 Questions

Questions are the way the system receives information about the state of the current domain problem once a consultation begins. For this purpose there exists a series of question

generators that reside in the frame reasoner, the demon logic, and the production rule system.

PROGRAM ENTRY \Rightarrow KB PACKAGE \Rightarrow ASCII FILE

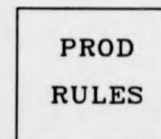
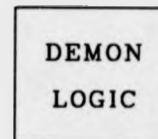
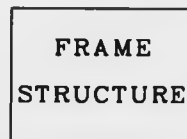
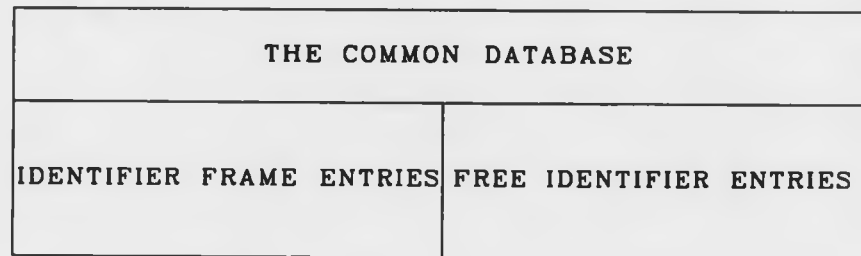


Figure 5.20 The Overall Database Entry Structure.

Canned questions sometimes exist in the QUESTION frame, and are indexed using the calling frame and associate slot

values. If there is no entry in the QUESTION frame then one is automatically generated. Four generators are employed by the expert system as follows:

- a) FASK - a simple asking routine used by the frame reasoner.
- b) FCASK - a constrained asking routine that is used by the frame reasoner.
- c) FDASK - a default asking routine that is activate by the frame reasoner each time a default value is requested. If no canned question is available FDASK generates its own question from the values of the frame and slot arguments.
- e) RASK - a question generator used by the production rule system and demon logic. This generator does not use any canned questions and formats the questions from a tuple that is passed to it by the inference engine. The form of the tuple is IDENTIFIER RELATION VALUE. This is formed into a positive question and posed to the user.

Canned questions are stored in a QUESTION frame in the same format as other special frames. The frame is identified as QUESTIONS-{domain}, the slots of the frame represent the calling frame, the facets represent the active slot of the calling frame, and finally the list associated with the facet represents the question. All questions are stored as a flat list and are unhyphenated and de-listed before being

presented to the user. Figure 5.21 shows some typical canned questions entered in the QUESTION frame.

```
Frame name: (QUESTIONS
(LAB-SET-UP (WAVELENGTH What is the radiation wavelength?)
            (MILLER-INDICES Enter set of miller indices?)
            (ARC-RANGE Enter rocking scan in arc secs?))
(LATTICE (SUBSTRATE What is the substrate for sample?)
         (LAYER Enter all material used in every layer?])
```

Figure 5.21 An Extract from Question Frame for X-ray Rocking Curve Domain.

5.8.1 Help

F1 can be used at any time to seek advice on the reasons for a question. The helps are stored on a frame slot basis in a special HELP frame. Like the commands, they are canned descriptions attached to a particular frame and slot.

5.8.2 How/Why

During or after a consultation, it is possible to see the reasoning of the system by selecting HOW and WHY options from the main tools of the expert system. HOW gives an account of the following stages in reasoning stating:

- a) Agenda tasks and their priority.
- b) Forward chaining with the frame reasoner.
- c) Questions generation and the user response.
- d) New FACTs added to the database.

- e) Procedural attachments when and where.
- f) If value is inherited and from where.
- g) When demon logic interrupts reasoning.
- h) When production rules take control.

HOW allows the user to see the system in operation giving all the reasoning used during a consultation.

WHY is more selective, and only gives an account of what the user requires. WHY generates a series of tasks executed by the system, and allows the user to choose from any. An inference tree is generated for that option which is displayed to the user.

5.9 System Operation

In operating the expert system, an initial query is set-up in the form of a prescribed set of questions that classifies the structure of the crystal under investigation (superlattice, heteroepilayer or multi quantum well). The selected structure becomes the current goal of the system, and the proof of the structure the first item on the agenda. If the goal is decomposable then the task controller will have "plans" for solving such a task and these will become an ordered list of sub-goals on the agenda, which in the case of a known structure would be simulation required or simulation not required. The Inference engine now tries to prove the first sub-goal which in the case of a heteroepilayer structure is more likely to be no_simulation. This becomes the current goal of the Inference engine and

the best hypothesis. Knowledge is sort to prove the goal using the knowledge base which in-turn may required inputs from the External procedures or User interface. In the case of a no_simulation sub-goal there is data driven procedure in the User interface for collecting structural information about the X-ray spectrum including questions on:

- a) The Thickness of the substrate between 0.5-5 microns,
- b) Two peaks on the Rocking Curve,
- c) Peaking splitting > three times width of larger peak.

In the case of the peak splitting ratio an external routine exists for calculation if required by the user. The other two questions can be answered from observations of the X-ray spectrum of the crystal sample. If all these assertions are true then it can be inferred that simulation is not required. This is added to the Database through instantiation of the TEST RESULTS frame, no simulation becomes the current hypothesis, and the task controller compares the database item(s) with the current hypothesis to check that sufficient evidence has been acquired to maintain it. If there is sufficient evidence then the next sub-goal is selected, but if not, then the no_simulation sub-goal is pursued cyclically until the probability of it being correct is sufficient to satisfy the task controller given the data discovered through the knowledge base. Once no-simulation is proved, the expert system tries to establish the structural parameters of the crystal without comparing the X-ray spectrum to a simulated output. This involves the

application of general equations for describing the parameters of the spectrum. The knowledge base contains general description of rocking curves from its LATTICE parameters which can be established through the application of constraints to the known structure of the crystal. The Inference engine maintains a current hypothesis describing the inferred structure of the crystal and attempts to prove this from a combination of matching user input to knowledge in the frame system.

When simulation becomes the sub-goal, the task of the system is to guide the user to iteratively produce closer and closer approximations between the experimental curve and the simulated curve. However, to solve this problem the task selector produces three new sub-sub-goals which divides the simulated curve into three areas: the substrate peak, layer(s) peak, and satellite peak(s). Proof of each of these tasks consists of altering the lattice parameters. The types of knowledge required to perform these operations are stored in the frame system and the slot values acquired from external sources in the External procedures. The parameters are changed according to the "closeness" of fit to the relevant section of curve, closeness as a measure of the probability of a match. As yet this measure has not been solved, but will hopefully be incorporated in the prototype version. Once proof of these sub-sub-goals is found, all tasks on the agenda are complete, the best hypothesis is selected and the structural parameters conveyed to the user. However, the task selector must be satisfied that the data

base items provide sufficient information since a match is never perfect and open to judgement.

5.10 Conclusions

The combining of frames with logic-based inference has proved to be an effective way of handling the open ended, yet structured domain of X-ray rocking curve analysis. The decomposition of the spectral output into a sub-goal structure has introduced a level of planning into the problem solving procedure. This has controlled the reasoning process and allowed the development of a separate question strategy which was an important consideration given the "training" function of the system. The next chapter will look at ways of building knowledge into the expert system core, and highlight the use of a novel knowledge elicitation methodology.

Chapter 6

Knowledge Elicitation and X-ray Rocking Analysis

6.0 Introduction

The acquisition of knowledge from the expert is central to the task of building expertise into an expert system. The knowledge engineer may be required to use any number of techniques to unlock the knowledge from the expert, and it has been pointed out by Wright and Ayton (1987) that the manner in which the knowledge is extracted from the expert needs to be formalised and structured in a way that it has not been in the recent past. The difficulty with understanding expertise is that by definition it is regarded as a skill, and a skill is something that is either partly or completely unconscious (Legge and Barber 1976). It is, therefore, important that the techniques used to elicit knowledge from the expert are valid, and make sense in light of the overall design of an expert system, maintaining the integrity of the information in the knowledge base.

This chapter is divided into three parts. Part I will examine the main knowledge elicitation techniques and Part II will outline the application of these techniques to the example domain of X-ray rocking curve analysis. Part III will outline a new knowledge elicitation technique; the application of the technique to the example domain, and the impact of this new knowledge on the design of the expert system. This technique was specifically designed to extract knowledge from the X-ray rocking curve domain which was then used in the building of an analogical reasoning system (See

Chapter 7). As a whole, the chapter will show how the knowledge elicitation affects and is affected by the design of a system. Here it is argued that both are interconnected and cannot be performed in isolation.

Part One

(Existing Knowledge Elicitation Techniques)

6.1 Knowledge Elicitation Methods

Broadly speaking there are five different ways of extracting information from the expert all of which can be used in conjunction when building an expert system knowledge base. These are as follows:

- a) Interview Techniques
- b) Protocol analysis
- c) Classification Techniques
- d) Goal Decomposition
- e) Machine Induction

Research has been conducted on providing a complete environment for knowledge elicitation using all of these techniques, and includes work at Boeing research centre on AQUINAS, an expert system transfer system (Bradshaw and Boose 1987), and KRITON a knowledge acquisition tool (Diederich, Ruhmann and May 1987). Techniques of this kind are currently available as an integrated knowledge elicitation environment called NEXTRA, and include:

interviewing methods, Multidimensional scaling, Repertory grids and Hierarchical clustering, Mapping, Induction, and Multiple Experts and Perspective Analysis (Neuro Data 1991). In this chapter these five techniques are present as two different approaches to the problem of knowledge acquisition: intra-personal methods, composed of interviews and protocol analysis; and abstractive methods, composed of classification, goal decomposition and machine induction. This distinction is made because the former methods rely on the expert being consciously aware of their expertise, whilst the later methods assume that a proportion of expertise is sub-conscious with the expert only becoming aware of it through an abstractive process.

6.1.1 Intra-personal methods

The techniques of interviews and protocol analysis are intra-personal methods, and they attempt to elicit knowledge from the expert by providing a framework for verbally expressing ideas stored internally. The success of this approach is dependant on the expertise being consciously accessible; the interviewer or knowledge engineer being able to assimilate the expertise, and the interviewee or expert being capable of communicating the expertise.

6.1.1.1 Interviews

The interview technique is the commonest method of extracting the information required to build an expert system, and probably the technique favoured by both expert

and knowledge engineer. It is a naturalistic method of the gaining expertise and enables the participants to freely exchange information in either a structured or unstructured manner. The interview can be in a number of forms and the three suggested here are:

- a) Informal Interview (unstructured) non-topic orientated
- b) Formal Interview (structured) topic orientated
- c) Seminar Presentation by Expert followed by questions.

There are also many different types of interview technique and different styles for extracting information, but all with the essential aim to initially gain an overview of the subject area from the expert either in the form of an introductory series of interviews with no fixed agenda, or in the form of the knowledge engineer becoming familiar with the problem domain followed by a series of confirmation interviews with the experts (Wellbank 1983). This process does not involve extreme detail and is aimed at giving the knowledge engineer ideas about the size and complexity of the problem domain.

Having outlined an overview of the problem domain, it is possible to divide the domain into topic areas with further supplementary interviews to extract information about each area. The focused interview is the general term used to describe this structured approach, with the knowledge engineer directing the interview procedure from an agenda.

The aim of these topic interviews is to explore the boundaries of each topic and relate these to the overall problem domain. Again, as with the introductory interview(s), extreme detail is avoided, the main aim being to obtain a 'glossary of technical terms and ideas' relevant to the domain.

Once the introductory and focusing strategies have been completed, the knowledge engineer considers the development of structured interviews. Structured interviews extract detailed knowledge on a chosen topic area. At present no literature exists suggesting the order in which topics might be explored, and there are no guide lines to say if, for example, the knowledge engineer should work with simple topics first followed by complex topics and vice versa, or central topics followed boundary topic and vice versa.

However, a number of techniques are available for exploring topics in this structured stage and these are referred to as probes. Probes aim to elicit knowledge from the expert in a systematic way and help in the structuring of the interview.

a) The Addition Probe - The knowledge engineer requests either directly or indirectly more information about a topic or sub-topic.

b) Reflecting Probe - The knowledge engineer summarises what the expert has said in order to allow the expert to further elaborate on the sub-topic.

c) Directive Probe - Used to shift the interview onto another level, thus the expert is either being too general or too detailed.

d) Change-of-mode Probe - This is used to provide another view point on the topic or sub-topic, thus to talk on an abstract level to one of examples or vice versa.

e) Defining Probe - The knowledge engineer may require the expert to explain the meaning of a specific concept. The probe is used to make explicit what re-definitions or definitions are required.

Probes are useful strategies for controlling the direction of the interview and are generally employed to extract details from the expert during the structured interview stage, although they can also be used in focusing.

6.1.1.2 Protocol Analysis

Protocol analysis is a technique used to extract the reasoning strategies of the expert. It involves the knowledge engineer making a detailed verbal and/or visual study of the expert in action, and then requesting the participant to explain their actions either concurrently or retrospectively. Through the use of a comprehensive and carefully sampled selection of typical and atypical examples of problems within the domain, it may be possible for the knowledge engineer to build a set of procedural processes used by the expert. Protocol analysis may reveal the

sequencing, selection and employment of declaratives within the domain, and consequently build a catalogue of heuristics to help in analysing the domain. By using a selection of example problems, it may be possible to shape the route through the declaratives and record the frequency with which certain routes and concepts are used. In this way protocol analysis provides supplementary information about the domain not necessarily expressed using formal interview procedure. Some researchers in the field of knowledge elicitation favour the use of retrospective protocols as they involve the expert in commenting on their behaviour after the event (Wellbank 1983). Concurrent protocol analysis is often difficult for the expert to perform as it is unlikely that they are familiar with commenting on their behaviour as they perform the task. Nisbett and Wilson (1977) go further than this in suggesting that protocols are just another form of introspection, and that the 'think aloud' techniques are invalid because you are asking the expert to tell you what they are thinking as they are thinking it. The authors believe that the expert will be able to say something during protocol analysis, but it will not be a valid representation of the thought process because it is impossible to express that information.

To combat some of these criticisms of protocol analysis it has been suggested that the knowledge engineer perform the task under the supervision of the expert with the expert pointing out deficiencies in the actions of the performer (Wood 1986). This technique may provide the knowledge engineer with an insight into the skills of the expert in a

way that is conscious. In this regard, it is often believed that skills are conscious and, thereby, communicatable during the learning process and, hence, easier to elicit.

6.1.1.3 Discussion

To a certain extent the interview methods enable the knowledge engineer to formalise some of the knowledge of the expert. The expert may have internalised some if not all of the expertise through verbal communication, and in this respect the same route may be tapped using re-communication techniques such as the interview. However, although this may be a valid way of eliciting knowledge, expertise may have been gained by other means such as the practical use of existing expertise, or through internal cognitions, or as the result of combining disparate sources of information not necessarily directly related to the domain of expertise. It may have taken many years to acquire the domain knowledge, and the dynamic process may have been the result of unique methods of selecting information relevant to the domain and rejecting information irrelevant to the domain. The static interpretation of the experts knowledge through interview technique may, therefore, fail to capture the history of the experts understanding. To this end, intra-personal methods may not always be the best method of extracting information about the problem domain, other forms of knowledge elicitation based on a cognitive strategy may be required.

6.1.2 Abstractive Methods

There are three abstractive methods frequently used to elicit knowledge. They are classification techniques, multidimensional scaling, and machine induction. These methods use indirect, often experimental procedures, to elicit knowledge from an expert. In this regard, the knowledge engineer is now no longer the interviewer, but the experimenter, and the expert is now no more the interviewee, but the subject. Furthermore, when knowledge engineers use abstractive methods they have a very different attitude to the process compared to when they use intra-personal methods. With the former approach knowledge elicitation is regarded very much a science, whilst in later situation knowledge elicitation tends to be viewed as an art.

Abstractive methods are a useful way of trying to elicit expertise since it is generally accepted that certain aspects of an experts performance are the result of unconscious and automatic cognitive processes (Legge and Barber 1976). From this perspective it can be seen that, even with the most skilled interviewer and the most detailed protocol analysis, it is not possible to extract all knowledge from the expert.

6.1.2.1 Classification

Classification techniques derive their current impetus from the cognitive approach that outlines ways in which knowledge is stored in memory. The main concerns of this area of psychology is in the structure of long term memory and through implication the possible routes used by the expert

in applying knowledge. The assumptions of these elicitation techniques are that by classifying the structures of knowledge from the expert it will be possible to duplicate the structure within the knowledge base of an expert system. The knowledge engineer can use a number of classification techniques to elicit knowledge from the expert including:

- a) Multi Dimensional Scaling (MDS)
- b) Concept Sorting

6.1.2.2 Multi Dimensional Scaling

MDS was first introduced as a technique for attempting to discover the internal structure of the human mind by Kelly (1955) as part of his personal construct theory. As applied to current methods of knowledge elicitation, MDS initially requires the expert to list important objects from the problem domain (this is similar to the intra-personal glossary). In the next stage the knowledge engineer randomly selects three of the objects from the expert's list and asks the expert to pair the most similar objects and explain the construct behind the pairing and behind the discrimination between the paired objects and the unpaired object. This process is repeated until all possible constructs are elicited, and then the knowledge engineer asks the expert to scale each objects against a bipolar axis for every construct. This process eventually forms a grid for the problem domain and through the use of factor analysis, the knowledge engineer is able to compare objects from the grid (Hart 1986). Scattergrams can be employed to statistically

analyse the relevance of the concept to the domain. This latter process is important since the use of MDS can throw-up false constructs through irrelevant pairing of objects. Gammack (1989) has devised a system for classifying objects within a problem domain using the MDS system, and introduces a validation technique for testing the psychological reality of the constructs defined by the classification methodology.

6.1.2.3 Concept Sorting

Concept sorting is essentially the same as MDS and also involves the structuring of object/concepts through grouping. It differs from MDS in that the technique is directly applied to relationships between objects and not a statistical relationship. In this classification scheme the expert is required to group objects according to concepts relevant to the domain, starting with large groupings followed by further detailed breakdowns of each group. The expert is then required to give typical examples of members of the category to enable the knowledge engineer to develop ideas about the features used by the expert to support the category. The knowledge engineer then presents a possible structure for the problem domain and this is verified by the expert. This structure is most likely to be in the form of a hierarchical system of relationships between objects.

Such classification techniques are especially useful if the domain has many objects contained within it, and if object orientated programming is to be used when developing an expert system.

6.1.2.4 Goal Decomposition

Classification techniques are useful means of understanding the declarative knowledge of the problem domain. However, these methods say nothing of the ways in which the knowledge is to be used. Procedural knowledge can be captured through the use of goal decomposition. Goal decomposition requires the knowledge engineer to illustrate a conclusion or a goal within the domain, and ask the expert to explain the tasks or conditions that have to be met in order for that state to be achieved. This process often results in the expert dividing problems into sub-problems and so on, thus articulating the sequencing and interrelationships of problem solving. Goal decomposition shows how the expert utilises the knowledge available.

The explanation of example conditions is a further extension of goal decomposition and again requires the expert to solve the condition. This technique is particularly helpful in situations where backward chaining is used extensively within the system.

6.1.2.5 Machine Induction

Machine induction involves the development of learning strategies which usefully operate within a narrow domain, and learn through both example and continuous data. Classification tasks tend to be the most successful area exploited by machine induction, and INDUCE is an example of such a program. The program was developed as a structured learning algorithm using a beam search for the

classification of soya bean diseases (Dietterick and Michalski).

Rada (1983) suggests that the automation of the acquisition of knowledge is helpful in circumstances where there is a bottle-neck during the elicitation phase. The attitude of researchers involved in machine learning is that knowledge acquisition is slow because domain expertise is gained over an extensive period of time, and the result of assimilating declarative knowledge. Knowledge acquisition is, thereby, not regarded as the collecting and correlating of facts, but rather the intake of facts over time which are then formulated into structures that implicitly reduce the search space of the domain. This is a generalisable approach to the acquisition of knowledge, and to implement such a strategy it is necessary to understand the processes involved in linking facts of the knowledge base.

The generalised knowledge acquisition tools tend to acquire knowledge in two stages:

a) Information gathering;

b) Iterative refinement of the knowledge base.

Information gathering is the process of constructing a declarative map of the knowledge base, consisting of facts or concepts known to the expert. The efficiency of the search space used to relate together the facts or concepts is increased in an iterative manner with the expert interacting with the domain. The expert assigns values to

events, with the system assigning its own values to states that cannot be rated by the expert. The iterative process continues until the expert is satisfied that the ascribed values produce viable hypotheses to example conditions. The benefit of machine induction is that the bottle-neck produced by the knowledge acquisition phase is reduced because such systems are able to characterise the knowledge base without the intervention of the knowledge engineer. Mole is such an example of a knowledge acquisition tool (Eshelman and McDermott 1988).

6.1.2.6 Discussion

The general consensus of opinion from the literature is that the knowledge elicitation phase of expert system design is a complex and not especially well understood area. Buchanan illustrates the often tacit nature of knowledge from the expert in extracts of dialogue between the expert chemist and knowledge engineer during the design of DENDRAL, a system for discovering the structure of unknown chemical compounds (Buchanan and Freigenbaun 1978). The experiences during the design phase of MYCIN, an expert system for the identification of diseases, also demonstrates how time consuming and error prone knowledge elicitation could be (Shortliffe 1976). TEIRESIAS was introduced to overcome the problem area of expert system comprehensibility, debugging and knowledge elicitation (Davis 1983). The system was an extension of MYCIN and aimed at reducing the role of the knowledge engineer by manning both the user interface and

the inference structure with meta-rules sophisticated enough to allow the expert to amend the knowledge base. A querying facility was added, and explanations of the systems conclusions made available to the expert. These facilities allowed the expert to modify the knowledge base through feedback from the system.

These conclusions and approaches to solving knowledge elicitation seem to point toward the development of abstractive methods rather than intra-personal methods, and in particular, the need to draw on paradigms that examine the cognitive structures of knowledge. This is supported by the suggestion that domain knowledge is unlikely to exist in isolation from more general perceptual abilities of the expert, and supports a more generalisable approach to knowledge structuring (Madni 1988). This does not preclude the uses of intra-personal methods since even at its most abstract there still has to be some kind of communication between knowledge engineer and expert to enable experimental data relevant to the domain to be produced.

Part Two

(Knowledge Elicitation of X-ray Rocking Curve Analysis)

6.2 The X-Ray Rocking Curve Analysis Interviews

The first interviews were carried out without any defined agenda, and without the knowledge engineer knowing any specific information from the domain. One expert was questioned by the knowledge engineer, and three interviews

were conducted in all. Each interview lasted one hour, being recorded on tape and then transcribed by the knowledge engineer (see Appendix 1). The interviews were unfocused and covered the general aspects of the domain. The following areas were covered:

- a) The purpose of the expert system.
- b) The types of rocking curve structures to be analysed.
- c) Conditions required for rocking curve simulation.
- d) The information required by the user to simulate.
- e) Rocking curve structures that break the rules.
- f) Study methods between experimental and simulated data.
- g) General terms assumed by expert requiring explanation.
- h) The types of users that will operate in expert system.

Lecture notes on MWQ and Epitaxial structures were analysed by the knowledge engineer to gain a understanding of rocking curve complexity (see Section 5.1). From this information a forward chaining expert system was built containing a set of production rules. This was presented to the expert for comments and amendments. The system aided the user in deciding when a rocking curve required simulation. The system also had a training function in giving detailed explanations for each question generated. This suggested that it was worth developing an expert system for X-ray rocking curve analysis (see Section 5.2) The main problems were the difficulty in adding new rules without restructuring the entire system, and lack of a formal structure to the domain.

6.2.1 General Analysis

Taking into account comments by the expert on the preliminary expert system, the design requirements were extended, the aim being to build a flexible expert system core into which knowledge could be added. The preliminary interviews were re-examined on the basis of these new aims, and additional material sort in the form of papers, lecture notes, two further interviews each lasting one hour, a demonstration of the simulation software using protocol analysis, and a goal decomposition session. Topics covered were:

- a) A taxonomy for rocking curves.
- b) Problem solving procedure for the simplest structure.
- c) A Step by step guide to simulation.
- d) The probability of faults in epitaxial layers.
- e) Hypothesis formation

The papers and lecture notes covered materials relating to epitaxial defects and conditions giving rise to them. Elicitation tended to isolate small sets of rules pointing to specific isolated condition(s). The goal decomposition session worked through an example of how an expert might solve a simple structure, from the setting up of the experiment through to the simulation of the experimental rocking curve. Expert three went through the process of setting-up an experiment and producing a rocking curve, and then analysing it. This highlighted the procedures used by experts, and illustrated the two staged process of

generating a single hypothesis, followed by iteration down to proof or disproof of the hypothesis. Protocol analysis was used in a demonstration conducted by expert three with the simulation software for modelling a rocking curve. This session, also showed the two staged process, and illustrated the proof procedures used by experts when identifying a structure. Following this, two structured interviews were conducted by the knowledge engineer with expert two. Of the two interviews, the first concentrated on the development of a taxonomy of rocking curve, splitting them into broadly defined categories and sub-categorises. The interview brought out the structured aspects of the domain. The second interview centred on the generation of hypotheses and showed that the expert tended to keep a single best hypothesis in mind when analysing the structure, and only abandon this if there was strong evidence accumulated against it. The results of all this analysis was an expert system core for X-ray rocking curve analysis as detailed in Chapter 5 (see Section 5.3 for cognitive analysis and Section 5.4 for Expert system details).

6.2.2 Detailed elicitation

This stage in elicitation consisted of three structured interviews, one with expert two, and two with expert three, covering all topics raised in previous five interviews. Each topic was explored in detail to generate the necessary rules for a prototype expert system. The first interview, with expert two, covered the taxonomy of rocking curves, sub-classifications, and super-ordinate groupings. The formation

of hypotheses was also re-analysed to determine what constituted the hypothesis, how it was supported and how it was refuted. The remaining two interviews with expert three looked at the simulation process and the procedures used by the expert to iterate down to solution. The types of mistakes users with differing levels of expertise might make was also examined, and so too were the scope of problems requiring a solution.

6.2.3 Discussion

Analysis of all transcripts of the eight interviews, and comparison of these to the protocols and goal composition sessions, suggested that not all the key elements of the domain knowledge had been elicited. Experts were often unable to express exactly how they formulated a hypothesis, or what the exact description of the rocking curve should be. There was, therefore, some uncertainty about both procedural and declarative aspects of the domain. What did appear to be critical was the manner in which certain key features of the experimental rocking curve guided the formation of the initial hypothesis, and how experience of past problems of a similar nature were used in iterating down to a solution. This latter point was particularly important when considering the fact that the final expert system would have to cope with the analysis of novel structures not encountered before. Particularly lacking from the elicitation were knowledge pointing to a generalisable structure to describe the rocking curve based on the visual appearance of the rocking curve, and also knowledge about

how to represented past examples in an accessible form to the expert system.

Part Three

An Experimental Knowledge Elicitation Procedure for Human Visual Pattern Recognition

6.3 A Experimental Feature Extraction Technique for Eliciting Key Knowledge from the X-ray Rocking Curve Analysis Domain

Part II indicates that key parts of knowledge are missing from the X-ray rocking curve analysis domain, and that existing methods do not extract this. More specifically, there is a strong visual element in the process of X-ray rocking curve analysis which has not been articulated. This is evident when the knowledge engineer tries to established how the experts form hypotheses.

To elicit this knowledge a feature extraction technique is put forward to isolate key features from the domain using an experimental technique used in the identification of prototype structures. The process of producing such prototypes is known as concept formation (Posner and Keele 1970). To develop the technique as a method for knowledge elicitation, two sets of experiments are conducted. In the first set of experiments, the technique is applied to data, in the form X-ray rocking curve plots, to see if the domain is suitable for elicitation. In the second set of

experiments the technique is re-applied to the data to extract selected features from the domain and, hence, elicit visual patterns from the expert. In this respect, it is felt that it is the use of these visual patterns by experts when analysing rocking curve data that enables them to quickly form an accurate hypothesis without the need to re-simulate the rocking curve (see Section 5.1.2).

6.3.1 Concept Formation

Cognitive psychologists have provided experimental data to suggest that given the limited capacity of human memory, prototype configurations representing typical instances of concepts, be they objects or ideas, are central to the notion of concept formation (Posner and Keele 1970). Concept formation is the process of internalising a series of different events such that a subject subsequently responds to them with the same label or action. In general, conclusions of research from this area supports the theory of prototypes as a means of storing typical instances of a concept rather than many examples of the same concept, and findings relevant to this thesis can be summarised thus:

- (1) When subjects are exposed to a range of distortions of an unknown prototype during training, they are, subsequently, more likely to recall the prototype than the distortions (Posner 1969). This suggests that subjects form "averaged" concepts from a range of

members of that concept, and that this schema is cognitively more accessible than the members of that group.

- (2) Low distortions of a prototype are better classified than high distortions of the same prototype (Homa and Vosburgh 1976). This implies that in terms of expert knowledge structure it may be better to produce many closely associated prototypes rather than fewer distant prototypes.
- (3) The larger the category size (3 -> 9) the more likely classifications are to be accurate (Homa, Sterling and Trepel 1981). This finding supports the previous suggestion for category size, but there may be an upper limit where a large number of prototypes may result in the misclassification of data.
- (4) Empirical prototypes rather than objective prototypes are a superior way of representing concepts (Breen and Schvaneveldt 1986). In this instance, an empirical prototype is a feature-averaged prototype that occupies the centre of a defined object concept, whereas an objective prototype is just the average of all the members of that group that are random distortions of the prototype.

As prototypical concepts are cognitively valid methods of knowledge representation, they are beginning to be used

7

within the expert systems framework. Recent work by Aikins shows how prototypes can be specified as part of the domain knowledge, and it indicates that they are a useful method for exploring general problem solving (Aikins 1983).

6.3.2 A Knowledge Elicitation Design using Concept Formation

Experimentally, the materials used to demonstrate concept formation usually consists of dot matrix patterns, or scattergrams, that have a neutral impact with regard to subject experience, organised around a prototypical average pattern (Figure 6.1).

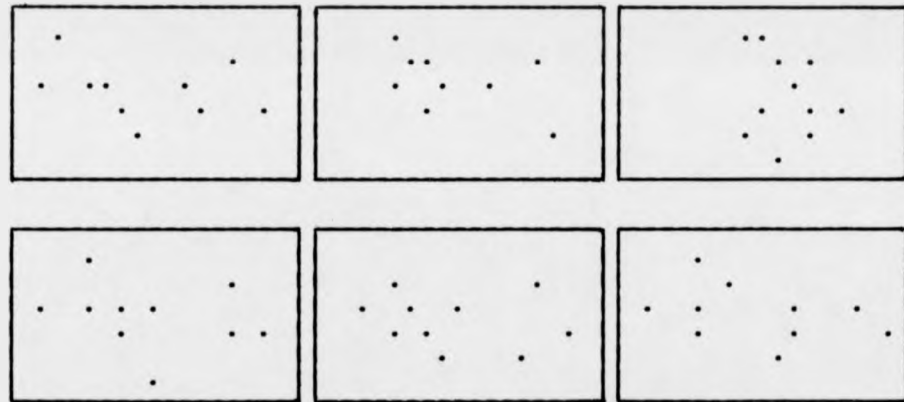


Figure 6.1 Top row shows three random dot prototypes. Bottom row shows three distortion, or transformations of the left prototype.

Depending on the aims of the experimenter, subjects may be exposed to a training set of prototype distortions, and then

required to either recall or categorise data in a second session. Data presented in the second session will consist of previously shown distortions belonging to the training set, new distortions, and the prototypes from which all distortions are generated. Concept formation is adjudged to have taken place if subjects are statistically more likely to recall, or correctly classify the unseen prototypes than the distortions shown in the training session. The assumptions of these techniques are that subjects do not possess the concept prior to the experiments, but that during training they form an internal schema that closely resembles the prototype, and that it is the subsequent use of this schema that increases the probability that they will mistakenly recall or correctly classify prototypes.

When designing a knowledge elicitation technique based on the procedures used in concept formation, it was initially assumed that experts possess a X-ray rocking curve schema, and that it is possible to match the closeness of experimental data to the experts' internal representation(s) by comparing theirs to that of a novice. The closer the match between the experimental prototype and the experts' schema, the higher the probability of prototype recognition when compared to the less knowledgeable subject groups. The usefulness of this approach is that by manipulating each subject groups (experts and novices) exposure to prototypes in an initial training session of prototype transformations, it is possible to statistically compare subject performance for recalling prototypes, and hypothesise that an increase in the probability of the experts recalling prototypes when

compared to novices indicates a structural similarity between the grouped training data and certain aspects of experts' schema. Each different structure or grouping of the training data could, therefore, be considered as a feature of the prototype, and hence a possible feature of the experts' schema. In developing such an experimental elicitation technique the following procedure is recommended:

- (1) Statistical design: Create a verifiable design to test for the existence of prototypes within the domain from which knowledge is to be elicited. This is the test domain. Decide on a control domain to achieve a base level performance for comparison with the test domain, and arrange the experimental conditions so that the framework isolates expertise, domain, and prototype performance. If the domain is verified then repeat the framework without controls, and use the prototype effect as an indicator for guiding the structuring of data. The stronger the prototype effects the closer the structure of the experimental data is to the schema.
- (2) Data production: Create a representative sample of prototypes from the domain and distort them to generate a pool of data. Sort the data into sets of features that other forms of elicitation suggest might be important in the domain and use these as training sets. Create further sets of data using data from the training set, data belonging to, but not included in the training set, and the prototype(s) from which the

data was generated, and use these as the experimental sets.

- (3) Subject groups: Create a minimum of two subject groups, one of the domain experts, the other of domain novices. Assume that experts already possess a schema, but that novices do not. An increasing prototype effect with expertise (subject groups) indicates a matching between schemas and experimental data organisation.
- (4) Data Presentation: Present data in paired sessions, the training set(s) followed by the experimental set(s). Decide on a training time for presenting training data (5-10 seconds per data item), enough to allow formation of iconic concepts, but not enough to allow elaborate feature analysis. Set an experimental time for testing (3-7 seconds per data item), a short enough period to induce mistakes in subject performance.
- (5) Experimental Procedure: Employ an experimental measure for recording subject performance after training. Use either recall rates, with subjects grouping experimental data shown following training as seen or unseen; or categorisation of data items, requiring subjects to group data into data blocks that reflected those shown in training. Train all subject groups with the same organised data sets, and follow each training session by an experimental session in which subject group performance is measured for prototype effects.

(6) Analysis of results: Evaluate prototype effects using an analysis of variance (ANOVA) (Fisher 1954), and probability scores (Plutchik 1974). The former technique describes the total variability across a design, and isolates the variability of factors within it. It assumes a normal distribution, that variances within subject groups are equal, and selection of data or subjects is random. It is ineffective at describing precise functional relationships between factors, but is useful for significance testing on components of variability. The later technique examines the individual elements of any significant factor. Probability score will describe the functional relationships between factors, but, unlike ANOVA, cannot indicate whether the results are significant. When analysing results look for a significant interaction between expertise and experimental performance, and then examine the comparative probability scores of subjects to detect prototype effects.

6.3.3 X-ray Rocking Curve Prototypes

As X-ray rocking curve analysis is a diagnostic rather than procedural problem, rocking curves have to be interpreted and, in the case of complex structures, matched to simulated models (see Section 5.1.2). This task would be easy if there were a limited pool of rocking curves to draw on, but there are an infinite number of possible structures, and new

materials and combinations of materials are being developed all the time. In this regard, the knowledge of the domain is far from static, and difficult to model. Prototype representations provide part of the answer by modelling graphical similarity, which is critical to the initial problem solving techniques used by experts.

A prototype rocking curve might consist of a feature averaged spectrum which takes into account the degree of peak separation in the sample between substrate and layer peaks, the number of satellite peaks, the overall intensity of each peak, and the width of each peak. However, without eliciting the knowledge from the experts it is not possible to know what features might be important to the experts in identifying and classifying rocking curves.

6.4 Procedural Knowledge and X-ray Rocking Curve Prototypes

The elicitation of knowledge from the domain indicates that when an expert performs X-ray rocking curve analysis the process is conducted in two stages (Tanner 1990). An initial guess is made by the expert as to the type of structure being examined from the plot of the X-ray rocking curve profile. This judgement is based on visual patterns existing in the curve. In the second stage, the expert simulates the curve, and iterates down to a solution that either matches or mismatches the hypothesis formed in the first stage (Tjahjadi and Bowen 1989). It has been found that the performance of experts and novices differs greatly in this respect, and that it is the utilisation of knowledge in the

initial hypothesis selection stage that is crucial. An expert will take on average 5-10 trials before discovering the structure reflected in a plot, whereas a novice may take up to 50 trials to achieve the same result. The contrast in performance between the expert and the novice, even with the same information available to both, implies that irrespective of the time taken between trials, the expert is better able to utilise the data, and hence more quickly arrive at the correct hypothesis. Performance is, therefore, the result of two factors:

- (1) The initial selection of an appropriate crystal structure.
- (2) The iterative procedure used to prove or disprove the hypothesis by altering lattice parameters and re-simulating the initial structure.

Given that only one crystal structure at a time is selected by both the expert and the novice during an analysis, the difference in performance may be partly due to the initial selection procedure used when forming a best hypothesis. In other words, the expert is more likely to select the correct best hypothesis first as compared to the novice.

A theory that states the difference in performance between an expert and a novice when analysing X-ray rocking curves is due to a difference in the ability to select the

relevant prototype. and that an expert has an effective schema that enables elaborate X-ray rocking curve analysis is, therefore, put forward.

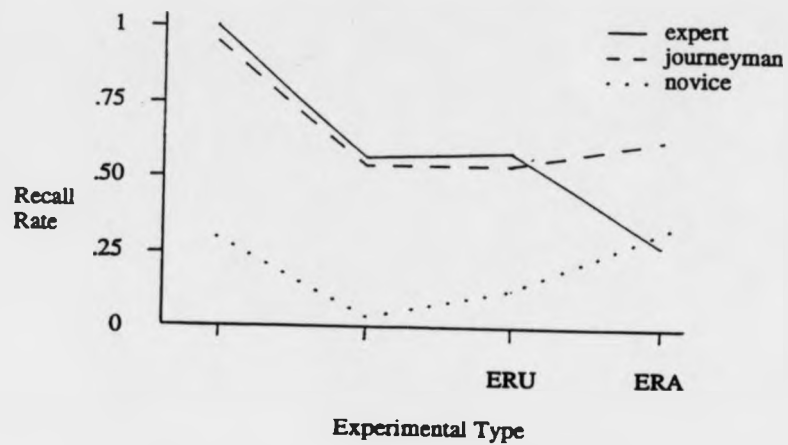
6.5 Method and Results of the Elicitation Process

Two sets of experiments are used to elicit knowledge from the experts. The aim of the first set of experiments is to test to see if prototype effects, and consequently schema can be isolated from the domain using expertise as the experimental variable. This is experimental Framework I. The aim of the second set of experiments is to identify the shape and form of these prototype structures from the domain by using the comparative performance criteria of experts and novices as the experimental variable. This is experimental Framework II.

6.5.1 Experimental Framework I

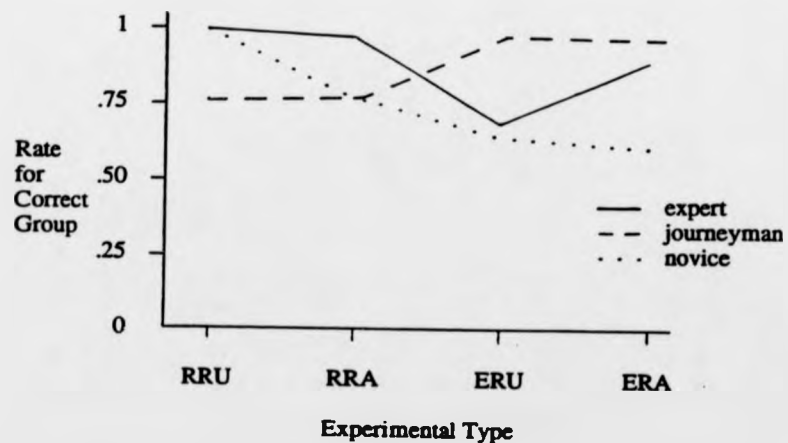
The first set of experiments shows the extent to which prototype effects operate within the X-ray rocking curve analysis domain using a designed based on Mill's method of difference (Plutchik 1974). The contrasting performance of experts, journeymen, and novices, is examined for prototype recall in two domains, the experimental condition in which expertise varied ie. the rocking curve domain, and a control condition in which expertise does not vary, i.e. the even function domain. All experiments have two sessions. In the first session subjects are trained on a random selection of data generated from the prototype structures. The data are

created by either randomly distorting prototypes, or distorting prototypes using rules. Random distortions use a specially designed algorithm to move each plotted point randomly within a prescribed area, which then applies a smoothing function to take out rough appearance of the plot. The rule-based distortions employed software to modify the values of the equations that generated the prototype plots before re-plotting them. These prototype structures are either standard X-ray rocking curve data (the experimental condition), or common even functions (the control condition). In a second session, the performance of subjects is tested following training. The performance of subjects is measured using both the correct rates of recall or successful categorisation of data. The data of the second session are composed of plots shown in the previous training session, plots generated from the same prototypes as the training data, but not shown in training, and the prototypes used to generate the data of session one. In all cases, the subject is never trained using the original prototypes, but only from the data generated through distortions of the prototypes. With perfect recall the subject should never register the prototypes as having been seen before. Figure 6.2 shows the probability of recalling prototypes for subject groups with differing levels of expertise for both the experimental condition and the control condition and Figure 6.3 shows the probability of correctly categorising the prototypes with the same subject groups (Henson and Tjahjadi 1991).



De=Domain Even, Dk=Domain X-ray RC, Ta=Random Transformation, Tu=Rule-based Transformation

Figure 6.2 Prototype Recall for Expertise



De=Domain Even, Dk=Domain X-ray RC, Ta=Random Transformation, Tu=Rule-based Transformation

Figure 6.3 Prototype Categorisation for Expertise

6.5.2 Experimental Framework II

In the second set of experiments, two sets of subjects, experts and novices, are tested using the X-ray rocking data generated for the first set of experiments. The same experimental format is employed expect that this time there is no control condition, and the training data are systematically selected from pool of prototype distortions. The training data are grouped into feature types and the selected feature held constant whilst other features remain variable for that group. Interviews conducted with experts (Bowen 1989, Tanner 1990) revealed that there were nine features that could be considered important in defining the overall shape of the rocking curve and, hence, contribute towards visual patterns used by experts in cognition:

- * Lattice Type
- * Number of Peaks
- * Peak Positions
 - Peak Height
 - Peak Half Width (width of peak at half its height)
- * Peak Density
 - Peak Integrated Intensity
 - Peak Shape
 - Peak Associations
 - Background peaks

The asterisked features have been selected for testing in Framework II on the basis of their generalisability. The other features either over-lap in function, are too

difficult to define or are not variable enough to be used as a means of defining rocking curves.

Lattice Type

The crystal lattice is made-up of a substrate upon which is grown one or more layers. Exposing the sample to X-rays using double X-ray diffractometer generates particular patterns that can be classified according to the structure and composition of the crystal lattice (see Table 5.1). These patterns suggest an overall shape to the rocking curve.

Number of Peaks

The number of peaks sometimes does not correspond to what is expected from the lattice. For example, two peaks lying close together along the X-axis may either cancel each other out or alter the true shape of the peak(s). Peaks can also interfere with each other and produce fringes around the shoulders of the peak(s). To isolate true peaks, therefore, a Poisson distribution is assumed for the rocking curve profile and the standard deviation (SD) of ± 3 set as an acceptance given the formula:

$$SD = (Ct) 0.5$$

where C is the count rate or X-ray diffraction for a given arc point and t is the counting time of that point (Bowen 1989). This formula is applied in a peak finding algorithm. It states that if a peak falls within the acceptance range

then it is considered significant. If the peak lies outside the acceptance range then it is merged with an adjoining peak, but if there are no adjoining peaks then eliminate it. The algorithm eliminates noise from the profile and cancels peaks that are not really present. Peak count is a quantitative measure, with any increases in the number of layers equating to an increase in peaks.

Peak Positions

Peak position refers to the location of peaks on the X-axis of the rocking curve profile. All locations are normalised around the substrate peak, which is always at zero arc seconds, and layer peaks are located positively or negatively along the axis of rotation in relation to the substrate peak by their maximum intensity at a given point. The exact centroid of the peak is found by fitting intensity ranges within 20% of peak height to a cubic polynomial and applying regression analysis (Bowen 1989). As peak position varies with the thickness of a layer and layer composition, it is used to determine the qualitative aspects of the sample.

Peak Density

This is a measure of the area under the curve within a given arc range. It is determined by integrating with respect to reflectivity and arc rotation between the lower and upper limits of the profile (Loxley 1990). The limits are 25% of the total arc spread either side of the substrate peak. This

measure in effect records the spread of information across the complete profile. Peak density is high when most of the information is centred around the substrate peak. It is lower when peaks appear some distance away from the substrate peak. This measure is determined by the qualitative aspects of the sample such as layer composition, curvature in the sample, layer mismatch, or defects in the lattice structure.

Table 6.1 shows the training data is split into four groups, each holding only one of the four selected features constant, and, thereby, cognitively assessible. The aim of this group training is to isolate prototype effects in expert subject group for the constant experimental feature, thereby, providing experimental evidence that the training set taps into the schema of the expert.

Table 6.1

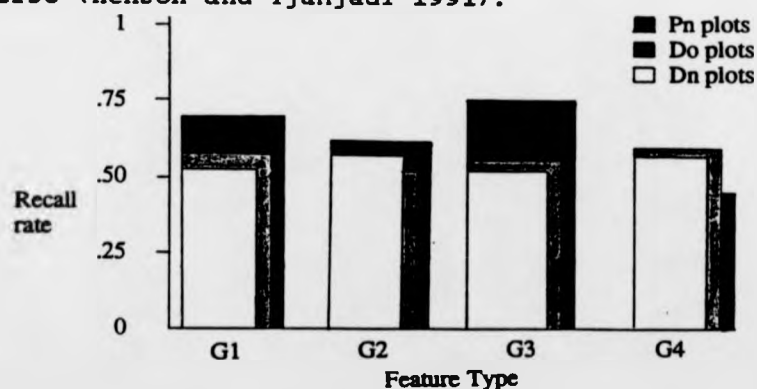
Classification of prototype distortions used in training

Feature	Group A	Group B	Group C	Group D
Peak Density	C	V	V	V
Peak Count	V	C	V	V
Peak Type	V	V	C	V
Peak Position	V	V	V	C

C = Constant

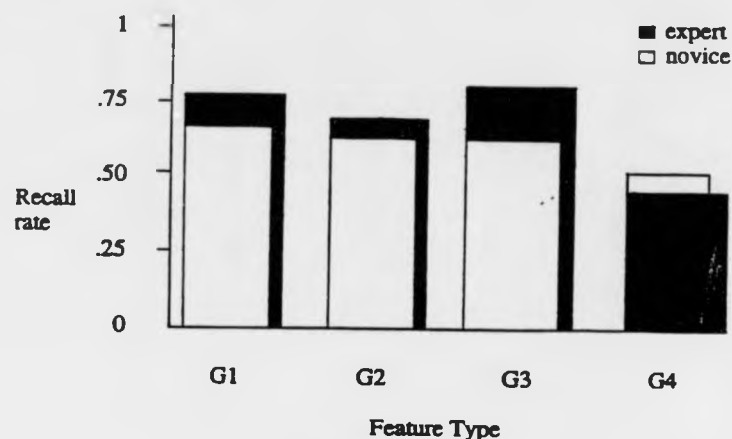
V = Variant

The experiment is performed on five experts and five novices as a randomised trial structure to statistically cancel learning effects across experimental groups. The training sessions, one for each feature, consist of showing each subject twenty eight transformations of a set of prototypes, and requesting subjects to remember each pattern shown. Following each training session is a recall session in which subjects are shown a selection a combination of transformation or plot types shown in training, a selection of transformation not shown in training, but belonging to the group, and the prototypes that generated all the transformations of that group. The task of subjects is to state whether the patterns shown the second session have been seen before. Figure 6.4 shows the probability of recalling each of the three plot types against the four feature groups, and figure 6.5 shows the same groups against expertise (Henson and Tjahjadi 1991).



G1 = peak density, G2 = peak count, G3 = peak type, G4 = Peak Position.

Figure 6.4 Recall of each plot type against feature type.



G1 = peak density, G2 = peak count, G3 = peak type, G4 = Peak Position.

Figure 6.5 Prototype recall against expertise.

6.5.3 Analysis of Results (see Appendix II and III)

Expertise is a significant factor of performance when recalling and categorising data. This effect is more under control during categorisation since it contributes to both the performance across domains and transformations. Experts tend to perform better in Framework I for their area of expertise, but this does not translate significantly to the control condition. This suggests that a domain rule effect may be in operation. Conversely, experts make more mistakes by recalling the unseen prototype data in their area of expertise, which indicates that they are using internal prototype structures. There seems to be some evidence for a meta-rule effect as experts perform significantly better in the rule based control condition for classifying plots. Rule

extraction of rocking curves helps performance in the control condition. In Framework II the concept formation technique is applied to the feature characteristics of the domain. The results reveal that training subjects on distortions selected for peak density, peak count, and rocking curve type produces significant prototype effects across expertise when recalling the data. This indicates that these three features are important in the formation of domain schema and are used by experts to help in the cognitive mapping of the rocking curve image.

6.6 The Impact of Knowledge Elicitation on the Expert system Design.

Knowledge elicitation suggests experts perform X-ray rocking curve analysis in two stages: firstly, the selection a hypothesis, and secondly iteration down to solution (see Section 6.4.2). Cognitive analysis of the domain indicates that under these circumstances a best hypothesis should be maintained during reasoning (see Section 5.3). To an extent the expert system core (see Section 5.4) reflects these needs by adequately representing a consultation in the following ways:

a) An overall consultation is hierarchically represented in a frame structure allowing the proof procedure to be systematic and abstractive.

b) Knowledge can be prioritised by the order of presentation within a frame and the order of the frame in the hierarchy.

c) Hierarchical inheritance and defaults are used in the frame system, both of which are computational cheap methods of running a consultation and gathering information.

d) Factual knowledge is stored separately from reasoning knowledge.

e) Non-structured elements of knowledge can be represented as procedural attachments, most typically as production rules, and called if required.

f) Constraints are added to user input and span the entire consultation through the use of a common database. This prunes the reasoning process dramatically.

g) Critical elements of knowledge can be represented as demons, interrupting the consultation and changing its direction.

h) Agenda are used to decide which tasks to solve. By arranging tasks in this way they can be sub-divided and stacked as required, and left unresolved until later in a consultation. All this gives the expert system flexible control.

i) Levels of control within the system can be changed to reflect the experience of groups of user. However, the user is only modelled statistically and feedback is not used to reflect the requirements of the consultation.

Unfortunately, there are a number of ways in which the core fails to represent the processes the expert uses whilst problem solving. These are especially reflected both in the mechanisms for selecting a best hypothesis, and the manner in which key features are used in this process. In this respect, the core lacks:

a) A way of configuring a hypothesis in a cognitively viable way. The consultation is neutral and simply follows the rules encoded without meta-level processing.

b) A way of representing past consultations, only representing the typical consultation. There is, therefore, no way of testing the probable success of a current hypothesis with previous ones that have been useful in the past. This makes the expert system inconsistent at the edges of domain knowledge.

c) The dynamic capacity to change the knowledge within. In this respect, there is no inbuilt mechanisms, aside of restructuring the frame system and production rules, for re-configuring knowledge over long periods of time so that configurations of features suggest a different direction in reasoning.

d) Ways of encoding the influences of key visual features on the consultation without continuous reference to them in the rule-base.

e) The ability to utilise user feedback when deciding how to formulate a hypothesis. Especially significant here is the ability or inability of a user to answer questions posed by the expert system.

To overcome these limits deeper knowledge is required, exhibiting some of the characteristics lacking in the expert system core. Such knowledge has been reflected in a number of recent expert systems including: Neomycin, a medical diagnosis system that used cooperative reasoning methods (Clancey and Letsinger 1981); OSM, a system for cancer diagnosis that uses meta-theories (Glowinski, O'Neil and Fox 1989); and an electronic trouble shooting system that uses reasoning from first principles and enables the relaxation of assumptions as necessary (Davis 1983). Deep knowledge attempts to deal with the issues raised by the field of human-computer interaction by accounting for the need to model a user (Madni 1988), making the consultation compatible with the end user conceptualisation of the task (Tjahjadi 1990), and by mimicking the reasoning and representations of the experts within the domain. With regard to the expert system for X-ray rocking curve analysis, there are a number of general design requirements that need to be specified and these are:

a) A method of representing the results of past consultations.

b) A way of configuring the expert system core to reference the past consultations.

c) A technique to match past consultations to the current consultation.

d) A technique for building general descriptions from key features.

e) A way of using key features to guide the consultation.

At its most general, the way to achieve these requirements is to tag existing knowledge in the core of the system with labels. Link the labelling into an inference mechanism, and build an inference structure to propagate down a scoring function for each key features, and use this to reference past examples. This takes the form of an inner expert system core linked to a conceptual database via an outer analogical reasoning cycle. The approach retains the existing expert system structure and the knowledge within and adds a separate module or inference engine as an outer core which links the inner core to a knowledge base of past examples.

6.7 Conclusions

Knowledge elicitation has proved to be critical to the success of the application of expert systems to specific problem domains, and current research in the area has demonstrated that this stage in the design and development of an expert system is both a complex and time consuming

exercise. A systematic approach to knowledge elicitation is necessary and this chapter has highlighted a number of existing techniques and approaches that can help in this task. Existing techniques have been applied to the X-ray rocking curve domain and the knowledge extracted added to an existing expert system shell originally designed using cognitive analysis techniques. The resulting expert system has proved marginally successful in solving problems from the domain. However, the system requires deep knowledge, and in order to partially meet this demand an experimental elicitation technique has been devised to extract subconscious visual patterns from experts. The technique has extracted three key features from the domain which are critical to the schema maintained internally by experts. Unlike other methods, this approach to elicitation also provides strong theoretical foundations grounded in conceptual formations from which an approach to expert systems design can grow. These formations emphasise the development and maintenance of prototypes, and the building and re-building of schema around existing cognitive structures. As yet these structures have not been represented within the expert system adequately. What is more, there is no way recalling the past except within the context of a typical consultation. These are problems that require deep knowledge solutions.

Chapter 7

The Development of Deep Reasoning and Representation in Expert Systems

7.0 Introduction

During the development of the expert system core for X-ray rocking curve analysis a combination of existing conceptual structures previously developed from within A.I. community were used to create an expert system shell. The cognitive analysis of the domain of X-ray rocking curve analysis served as a design guide. The outcome of this process was a combined frame based and production rule system controlled through the operation of an agenda and common factual database (see Chapter 5 Part Three). Building on this development, existing methods for knowledge elicitation were investigated (see Chapter 6 Part One), and then applied to the chosen domain (see Chapter 6 Part Two). Knowledge extracted from experts was represented in the knowledge base, producing an expert system for X-ray rocking curve analysis. The end-product was a working expert system, but one without deep reasoning. What the system lacked was a viable cognitive representation of the problem domain, and in particular the iconic representations of rocking curves. Further elicitation using a novel feature extraction methodology, based on the work of concept formation theorists, demonstrated the abstractive qualities of rocking curves, and showed that experts store schemata internally and use key features from rocking curves to map large visual

descriptions of existing problems to stored representations (see Chapter 6 Part Three).

This chapter will show how deep knowledge can be included in the expert system framework by re-defining the meaning of a consultation in deep terms. It will look at competing systems of deep representation and reasoning and assess the value of these to rocking curve analysis. Emphasis will be placed on the notion of analogical reasoning which is frequently used in everyday reasoning, and also by experts involved in X-ray rocking curve analysis as the following extract from elicitation data shows:

R.Henson. "Could an expert use information about one crystal type i.e. the rocking curve produced from one of those types you've mentioned to help solve another type?".

B.Tanner. "Oh yes"

R.Henson. "So if you are working in the dark you could reference a previous example rocking curve to help you solve another problem?"

B.Tanner. "Yes"

The outcome of this process will be an expert system core with deep knowledge encoded into its structure.

7.1 Deep Knowledge

The majority of expert systems developed to date have not tended to mimic the cognitive processes of the expert, but rather produce results that simulate the behaviour of the expert when problem solving. In other words, such expert systems make no attempt to represent the problem method and representations of the expert and treat such processes as a "black box". This might not seem directly relevant to many domain specific problems, but if a system is to be considered an expert system, then it is necessary for it to perform in an expert way and not just an algorithmic way. Most of the early expert systems used methods of representation and reasoning far removed from the methods understood to be used by experts, and consequently the performance of such systems, whilst promising in very limited domains, tended to be both inflexible and difficult to adapt as the need to develop Theriasis (Davis 1983) for the Mycin project showed (Shortliffe 1976).

In a recent conference on expert systems in engineering applications, most of the papers presented used a non-cognitive methodology in constructing the expert system framework (Tjahjadi and Bowen 1989). However, there was a general call for the development of second generation expert systems, which could be said to be closely allied with the field of cognitive psychology. Critical to this approach to solving problems is the use of analogy, which in general terms, is the application of known knowledge to a novel situation in order to generate a solution. The advantages of such an approach is that it both provides a means of

restricting search by analogical constraints, and enables a system to reason beyond its current knowledge. Such characteristics have the potential for learning and, thereby, modifying the knowledge base without the intervention of a knowledge engineer. This capacity for 'self improvement' provides justification for the development of such systems in engineering applications, since it is often the long term costs of maintaining a system rather than developing a system that proves to be the overriding difficulty. This approach fulfils the requirements of deep system for X-ray rocking curve analysis (see Section 6.6), and will, therefore, be adopted as the main emphasis for the design of a 'deep' expert system.

7.1.1 What Constitutes Deep Knowledge in Expert Systems

It has been suggested elsewhere that the development of deep structures within expert system are essential to the growth of research in expert system technology (Price and Lee 1988). The definition of what constitutes a deep system is everything that first generation expert systems are not and this includes:

- a) Flexible problem solving
- b) Good man-machine interfaces
- c) Good maintainability

7.1.1.1 Problem Solving

By implication a definition of flexible problem solving is the ability to change the plan of action to suit the current problem (see Section 2.3). Such specificity overcomes the boundary problems expert systems face when dealing with atypical problems, because the problem-solver does not assume typicality. Moreover, small parts of the problem solving strategy may contain peripheral knowledge even though the problem as a whole might be typical. It is, therefore, unlikely that an expert system built around the notion of typicality alone will perform expertly. The expert system core built for X-ray rocking curve analysis describes the consultation process in terms of a frame hierarchy (see Section 5.4.1.1). This frame structure contains frames each of which is a typical instances of an object. The complete hierarchy, thereby, represents the typical consultation. Special procedures have been built to cope with non-standard problems, but these have been elicited from the domain and are not part of the general problem solving procedure. This limitation needs to be overcome in order to develop a deep problem-solver.

7.1.1.2 Interfacing

According to the deep criteria, expert systems are only expert if they interact with the user in an 'intelligent' or deep manner: permitting the user to withdraw previous responses; hold records of past consultations (the historical perspective), allow voluntary information by the

user, give adequate reasons for the conclusions they reach, and tailor questions to the user's need. To a certain extent these requirements have been reflected by research into user modelling, which entails the building of a model of the kind of users operating the system, and altering the processes to accommodate them (Rich 1983). Systems such as 3M (Tahjahdi 1991) and WEST (Burton and Brown 1979) are user modelling sub-systems specific to a particular expert system, and tackle the issues of user expertise, and the teaching performance of expert systems. Running along side these developments has been the investigation of non-monotonic logic contribution to systems of truth maintenance (Doyle 1979). Such logic partly addresses the issue of flexible user responses to questions, permitting the user to violate previous truths by the maintenance of multiple hypotheses, although this is at considerable processing and storage expense (see Section 4.3.1). There are expert systems that develop user interfaces from this perspective, but they are limited in number and not currently viable for the reasons just stated. Database systems can help with storage problems by holding knowledge in compressed forms. A great deal of work is being conducted in this area and being applied to large databases (Kerry 1990). Interfacing of this kind is particularly important to the deep expert system because it allows the accessing of data from non-expert system sources, and provides a means of storing the results of consultations conducted with expert systems. This solves the historical problems associated with knowledge bases (Mylopoulos and Brodie 1991).

Some of these requirements are already included in the expert system for X-ray rocking curve analysis. Users can volunteer data by querying individual frames without engaging the rest of the hierarchy. However, as yet, there is no links to object databases and consequently no means of storing historical data, particularly those of past consultations. This is an important aspect of deep reasoning because it provides a means of comparison between the typical consultation expressed within the frame hierarchy and stored in the common database (see Section 5.7), and historical data stored in an object database.

7.1.1.3 System Maintainability

This aspect of expert systems has two areas of concern. Firstly, there are the practicalities of adding knowledge to what often becomes an increasingly large and complex knowledge base. Secondly, there is the issue of altering the existing knowledge to accommodate the needs of the ongoing consultative process. To overcome the problems of adding new knowledge, a knowledge acquisition module can be developed, allowing the knowledge engineer and/or expert to added new knowledge without needing to take note of the internal mechanisms of the expert system. This module should contain verification routines to ensure the integrity of the knowledge base. This problem has been solved for production rule systems by using a polymorphic approach (Yen and Jnang 1991). Such routines search for redundancy in the rules by detecting rule duplication, highlight rule clashes where

there are explicit contradictions, and report possible syntactic errors within the rules themselves. Modifying an existing knowledge base over time is a more complex task. It concerns the expression of change in the knowledge base over a long consultation period without the intervention of the user. This is a process that is natural to the human expert, and in itself results in expertise. However, the extent to which machines can modify knowledge and perhaps discover new knowledge has to currently be restricted to formal domains with definite proofs (Harmelem and Bundy 1989).

The expert system for X-ray rocking curve analysis has a primitive knowledge acquisition module, but this does not permit advanced monitoring of the knowledge base. Adding such sophistication to the acquisition module will, therefore, be a consideration. Improving the design development of a modifying structure for the expert system core is more important since the range of problems to be solved by the system are vast and everchanging (see Section 5.3.6). Without the automatic continual modification, or at least a built-in design potential for dynamic operation of the system, it may not be practical to develop an expert system from existing knowledge. This will, therefore, be a priority when building depth into the expert system core.

Form the general failing of first generation expert systems and the implications for deep systems that flow from this, a list of guide-lines for the development of a deep expert system are put forward. These recommendations emphasize the

accountability of expert system design to the domain they attempt to model, suggesting that:

- a) Knowledge elicited for the system is cognitively viable.
- b) The design of the expert system has a cognitively viable structure.
- c) The consultation process reflects the objectives of the expert system.

7.1.1.4 Knowledge Elicitation

Elicitation from the X-ray rocking curve domain was initially unguided by the design of the system and did not attempt to reflect the cognitive structure of the expert. However, as elicitation progressed a design for the basic system emerged, and it was at this point that the design of the expert system began to be reflected in the continued elicitation process. This has been referred to as knowledge assimilation rather than acquisition (Lefkowitz and Lesser 1988), and the degree to which new knowledge can be assimilated is a test of the cognitive viability of knowledge base structure. To this end, as well as using the standard elicitation procedures to build the knowledge base, a cognitively viable elicitation procedure was created to extract key features from the domain (see Section 6.3).

7.1.1.5 The Design Process

The design of an expert system can take a number of forms. This includes the creation of specific techniques, general techniques, or the application of existing techniques to solve domain problems. The nature of the resultant expert system is dependant on which options are chosen and in what order. The application of existing techniques will tend to support an engineering approach to the design, fitting together components to produce a working machine. The development of novel and general techniques will create a integrated design which matches the domain either closely or not so closely depending on the choice. The initial approach adopted during the design of the expert system core was to apply existing techniques to the domain through cognitive analysis. This is considered viable since it matches the domain requirements to established and 'approved' methods developed from within the A.I. research field. However, elicitation of the domain showed knowledge gaps, indicating that the initial design did not fit the domain in all areas. The design lacked a formal method for representing both key features, and past consultations. To fit the 'machine' to the domain, therefore, requires a specific domain approach, filling gaps in the knowledge. This approach adheres to model-based prototyping model, which borrows from the techniques of rapid prototyping (Hayes-Roth, Waterman and Lenat 1983), and concept modelling (Schreiber, Bredeweg, Dauoodim and Wielinga 1987). A general domain approach has not been adopted since this requires the expert system core to be completely re-designed.

7.1.1.6 Consultation process

Clancey (1989) argues that the functional cycle of the expert system is of overriding importance. The aims of model building are emphasized, giving rise to the notion of a knowledge base as a model of the world and the inference engine as a manipulator of that model. For example, medical diagnosis can be viewed as a patient->disease->therapy inference structure. This gives a perspective to the expert system builder, but is it the best available? Here it is argued that the role prescribed for the expert system as a problem solver affects fundamentally both the elicitation and design process and ultimately the final product itself. A more general inference structure for medical diagnosis might be monitor->diagnose->modify. In this respect, the aims of the expert system are now to describe the body as a whole system, diagnose malfunctions, and return the body to a stable state, rather than describe the symptoms of the patient, identify the disease, and prescribe a cure. This shows that the choice of the model is as important as the design itself.

7.1.2 Models of Input for a Deep Design Approach

Cognitive psychology provides a rich source of models from which to develop a deep design approach. They often bring a information processing approach human thinking, which is empirically based. Although much of the research is based on general cognitive abilities, certain aspects of research are considered relevant to specialised or expert thinking. Three

types of models that are of particular interest to expert systems design are those of analogical reasoning, exemplar based learning, and memory models.

7.1.2.1 Learning by examples

Dynamic models of expertise can incorporate a learning component into the structure, and the most general of these models is 'learning by example'. The other main methods are 'learning through actions' and 'learning through instruction' both of which are discussed in Chapter 2. The ways in which these methods of learning can be and these include: learning by analogy, generalisation, and discovery (see Section 2.3.4). Learning by examples has been explored most effectively in the visual field through the work of Winston (1975). The process involves identifying common structural features in the current problem and matching these to past examples. Examples are like cueing structures that map to the present problem through the relations that exist between the parts. Most importantly, the Winston programs learnt through both examples and counter-examples, thereby constraining the problem to a limited range of possibilities.

Within the X-ray rocking curve domain, examples are used by the expert to help them when analysing a rocking curve plot. Elicitation, suggests that the expert uses a prototype structure to give shape to the rocking curve, and compare new plots with old plots through a the analysis of features. In this regard, the iteration down to solution appears to

be a building process, matching with ever increasing accuracy the current problem to a selected prototype, until either a mismatch occurs or featural equivalence is achieved.

7.1.2.2 Memory Structures

Memory structures are not passive media of storage, but dynamic structures. There are a number of models that have been described in the literature and of these the dichotomous Long Term memory (LTM) and Short Term Memory (STM) theory, or duplex theory is most widely accepted (Klatzky 1975). STM is considered to be the working memory, with a limited capacity, and recall span. Research into the nature of this storage system suggests that it is a sensory register that has a limited capacity to apply semantic codes to information. LTM has been shown to be considerably more complex, and there are many differing theories as to its structure. All agree, that information is semantically encoded, and that it is probably hierarchically structured. Set theories have been proposed (Meyer 1970), along with semantic-feature schemes (Smith, Shoben and Rips 1974). Memory models are of interest to the expert system designer because they are dynamic storage systems, and this is important if long term changes in the consultation cycle are to be registered. Passive storage methods are built to remain static, memory models are designed to be in flux. As the domain of X-ray rocking curve analysis is constantly

changes, such models are an important input into the design process of a deep expert system.

7.1.2.3 Analogy

Analogical reasoning is a broad based problem solving strategy. It is recognised as the differentiation of target and source material within a computer system using a memory structure that defines the target as a problem interpreted in STM, and the source as a possible route to solution recalled from LTM. The process of analogical reasoning can be defined in five stages: (Hall 1989)

- a) Selection - selecting the most important experimental data from the target i.e. that which varies most from the expected average. For example, if the dog has no tail and this is an atypical feature then focus on tails. In the case of rocking curve analysis this might require prototypical descriptions for different types of curve and a comparative mechanism for judging that which varies most from the prototype.
- b) Recognition - recognising a candidate analogous source, from an target description. This would be based on the characteristics of the target.
- c) Elaboration - the mapping between source and target domains with the inclusion of inferencing methods.

- d) Evaluation - a system of choosing the best hypothesis with regard to its application to solving the problem. This would involve gathering evidence to support or disprove a hypothesis and might drive the questioning method to gather the most important evidence.
- e) Consolidation - the recording of results of the process as part of the learning paradigm. This would require the updating of LTM based on evaluation and involve adjusting typical examples within the cognitive map, creating a new source, or deleting a old source.

In operating these procedures, a source is identified by which the target can be compared, and any additional information attached to the source not identified with the target be mapped onto the current problem in a process often referred to as conjecture (Eskridge 1989). This involves replacing conjectured objects from the source with the equivalent objects associated with the target domain, and asserting them to the target. The new knowledge attached to the target is then evaluated assessing the current goal, and if necessary setting new goals. Figure 7.1 shows the basic elements of analogical reasoning adapted from Eskridge (1989).

As stated earlier, analogical reasoning is used by experts in X-ray rocking curve analysis. This makes it an area of direct relevance to the expert system design.

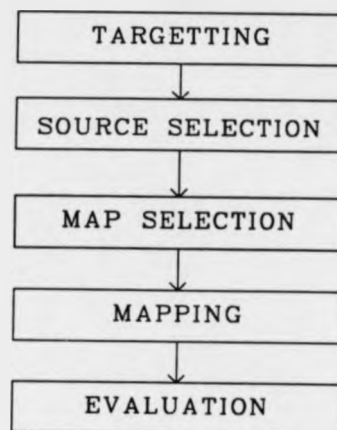


Figure 7.1 The Analogical Reasoning Process

7.2 An Expert System Model for Deep Reasoning

In order to implement a system for deep reasoning it is necessary to re-define the design of the expert system. This requires the re-examination of not only the parts that make-up the expert system, but the complete expert system and the consultation cycle in which it is embedded, including both the short term aims of the system as a consultative mechanism, and its role as a dynamic entity. There are a number of cognitive models that provide impetus to the design of the deep expert system (see Section 7.1.2). Of design interest to the development of a deep expert system are:

a) The configuration of the expert system architecture into STM and LTM modules.

b) The definition of tasks within the expert system in terms of the stages toward analogy.

c) The conceptualising of a database of source material for use by the expert system.

d) The maintenance of a historical perspective for past consultations.

7.2.1 A Re-definition of Problem Solving

The traditional view of cognitive activity is that it is a bottom-up process from which basic input patterns are matched and then categorised in a increasing abstract hierarchy (Klatzky 1975:141). The defining of schemata and the cognitive map in which they are embedded provides an alternative way of viewing cognitive activity, and one which operates through a cyclic interaction with the environment (Neisser 1976). The cognitive map represents what is often referred to as an orientating schema which is an information seeking structure that accepts information and directs actions. A schema is a basic cognitive unit that is totally internal to the individual and accepts information within its constraints. Larger schema seek further information and smaller schema are often embedded in the larger ones. All schema are capable of being modified by the environment

which gives this perpetual unit dynamic qualities. Frames are a computational interpretation of this concept and constitute an attempt to recognise the role of context and meaning in cognitive activity (see Section 4.2.3). However, the implementation of the frame structure has often failed to exploit the dynamic qualities of these mechanisms, embedding them in a static knowledge base.

7.2.2 A Model of a Consultation

Figure 7.2 illustrates this section's interpretation of the frame concept in respect to expert systems, and a possible development tract for the design of the system architecture with a learning component based on analogical reasoning.

In Figure 7.2 the expert system is divided into three segments: firstly, the outside environment which may include experiences not encoded into the expert system and the responses of the user; secondly, the knowledge structure of the expert system which retains a Long Term Memory (LTM) or knowledge base, Short Term Memory (STM) and procedures for induction; and thirdly, the action plans for moving forward in the environment including reasoning methods and planners for structuring the collection of data. The system is also divided into two circles which represent different levels of operation over time: the outer circle represents the complete processing structure of the expert system including its interactions within the working environment; the inner circle represents the individual problem solving sequences that over time alter the outer representation.

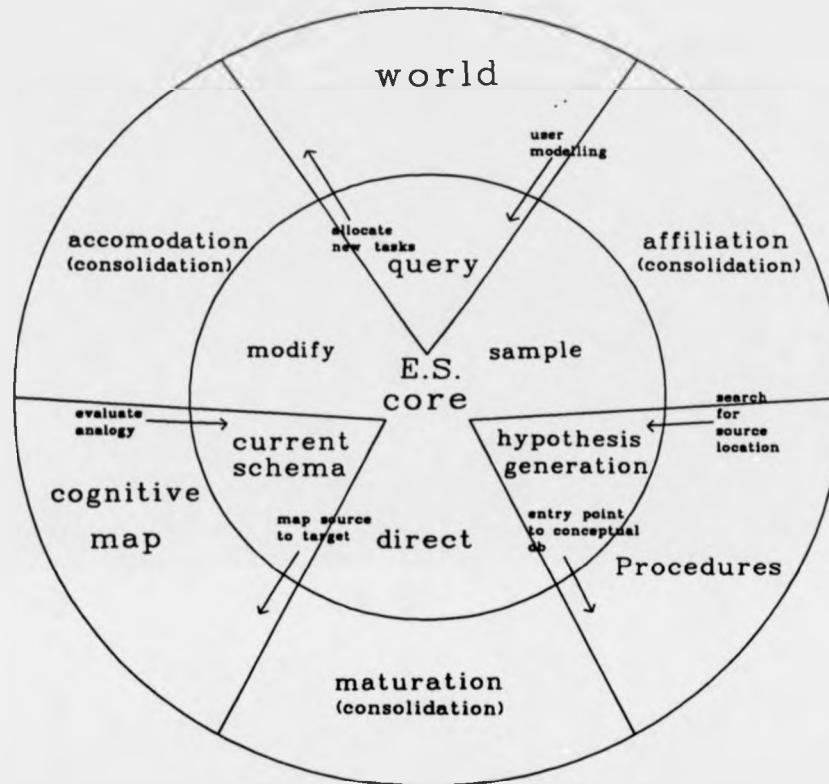


Figure 7.2 Overall Expert System Learning Structure

At the problem level, each time a query is set-up there is a possibility of the current schema (set of frames) stored in long term memory being updated. Thus there is a purpose to each transaction. The schema directs the inferencing by organising the requirements to satisfy the current problem. A current hypothesis is maintained using inference which

deduces new facts or rules from the knowledge base. Samples are taken from the real world during the process. New requirements are set-out in the form of problem statements from the real world which update the current query and these in-turn change the parameters for the operation of the schema. The perpetual cycle continues until a problem is resolved. At the domain level, learning takes place as a result of resolving problems, and at the end of each transaction the weighting between schema in the map is changed. This process is known as accommodation, a concept first used by Piaget in the description of cognitive development (Piaget and Inhelder 1977). This is the modification the internal representations, which in this case are structured as a cognitive map.

The fitting of reasoning to knowledge is a further long term change that takes place in the outer circle. This process is known as maturation, which in the context of the proposed model refers to the matching process of typical examples or prototypes with their associations ie., schema to the reasoning strategy or plans based on past successful reasoning strategy. This could involve either the updating of the reasoning plans or the addition, subtraction or re-assignment of schema to existing plans. The final long term change is that of affiliation which is the alignment of reason to the possibilities beyond the workings of the expert system. This may involve widening the sample space for a particular problem and possibly relaxing constraints place on the data, or it could mean restricting the sample space and tightening constraints on data from the

environment. Learning is the result of assessing the relationships between the three components in the conceptual model, and transforming the components of the system with a well defined set of operators.

7.2.3 How the Core fits to the proposed Deep Model

The expert system core as it exists (see Section 5.4), partially fits the model proposed in Figure 7.2. The actions of the expert system match the inner cycle of the model (modify->direct->sample) through the instantiation of the frame structure, the operation of rules of inference (demon logic, propositional logic, and constraints), and the seeking of knowledge by the posing of questions. The model has three components the world, the cognitive net, and the inference mechanisms. In the present system these are represented by the knowledge base, the inference engines, and user interaction along with the facts established in the common database. However, there is no equivalent process for the outer cycle of the model (accomodation->maturation->affiliation) since the core neither modifies the structure of the knowledge base, alters the inferencing methodology, or changes the structure by which data is stored over a long term period. Table 7.1 summarises the structural and procedural equivalences between the model and the core. As seen from Table 7.1, the expert system core adequately copes with the inner cycle of the deep model. Unfortunately, there is no dynamic means of altering the knowledge, inferencing methods, and the world model.

Table 7.1
Structural and Procedural Equivalences between the Expert
System Core and the Deep Model

Core	Model
PARTS	
Agenda	Algorithmic Reasoning
Dictionary	Cognitive Net
Common Database	Actual World
Frames	Cognitive Net
Production Rules	Algorithmic Reasoning
Demon Rules	Algorithmic Reasoning
Question Generators	Actual World
OPERATIONS	
Query	Query
Task Allocation	Direct
Demon Logic	Direct
Propositional Logic	Modify
Default Logic	Modify
Inheritance	Modify
Procedural Attachment	Modify

The aim is, therefore, to put forward a mechanism for meeting this demand whilst still retaining the existing expert system core.

7.3 The Implementation of the Deep Expert System

In viewing the outer cycle of model in Figure 7.2, an outer analogical reasoning cycle surrounding the core is proposed as a suitable structure to deepen the expert system. A conceptual database is proposed for the cognitive net, an analogical reasoning algorithm proposed for the inferencing structure and an evaluation program proposed for assessing accuracy of the information being received by the expert system from the world. Analogical reasoning has been chosen because of its capacity to operate dynamically through the process of consolidation and its suitability for mapping knowledge from outside the domain.

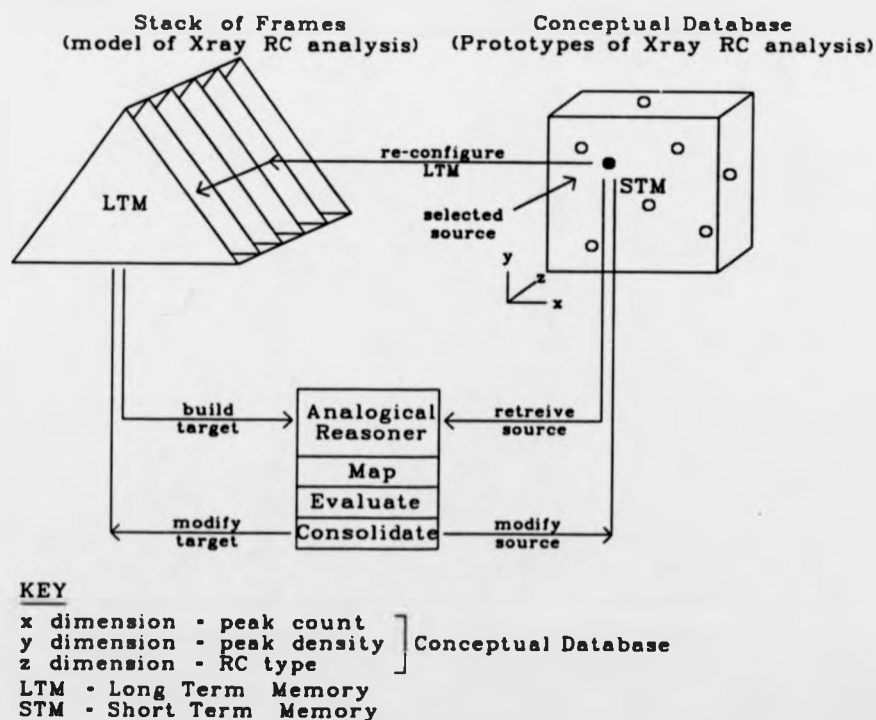


Figure 7.3 Schematic of a Deep Expert System Architecture

Under these circumstances the core acts as the typical problem or target description builder, the conceptual database represents the source material, and the analogical algorithm provides a means of mapping the source to the target and then assessing the value of the process. Figure 7.3 shows the architecture.

7.3.1 The Conceptual Database

The most widely recognised way for representing data conceptually within an expert system framework is through an object orientated database such as STATIS (Symbolics incorp.) or INTERNEST (Texas Instruments). These express both the declarative and procedural aspects of an object in the same manner as the frame structure. The difference is that each object is hierarchically free and the relationships between objects described through high level programming functions which serve client programs. CLOS is a typical programming standard used for this purpose (Keene 1990). This method was rejected on the grounds that it would either be necessary to conform to the object orientated programming procedures within the frames system or generate an interface between the expert system core and the object database. Furthermore, it would be difficult to apply the learning paradigm to such a system through analogical reasoning because relationships are expressed symbolically, and cognitive analysis of the domain indicated that it would be necessary to represent uncertainty explicitly, probably through statistical reasoning.

7.3.1.1 Representation

To store conceptual data, a multidimensional probability database is proposed. This involves the mapping of features on to a conceptual space. Each dimension is a probability vector in which all values lie between 0 and 1. Each dimension has a label that describes a feature and each dimension is independent of any others. The dimensions are divided into equal portions of probability so that incremental points of probability can be specified. Objects can be stored at these specified locations and, thereby, described in terms of probability. For example, a two dimensional space with labels: Peak-Asymmetry and Peak-Complexity; and objects A, B and C might be represented as:

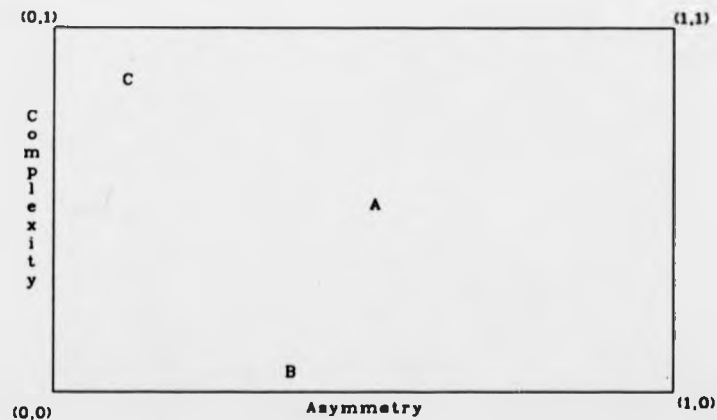


Figure 7.4 Representation of a Two Dimensional Probability Matrix for complexity and peak-asymmetry.

From Figure 7.4 the reading of the matrix shows three objects, each with a different probability rating for complexity and asymmetry. Object "A" has a rating of (0.5,0.5), "B" is (0.4,0.1), and "C" is (0.1,0.9). In descriptive terms "A" translates to an object that is equally likely to have peak asymmetry or peak symmetry, and equally likely to be a complex or simple in appearance. Object "B" can be described as very likely a simple peak with the possibility a symmetry. Object "C" is very likely to be a symmetric, complex peak.

In such a probability space, as the number of dimensions grows the descriptions they support become more complex. This is good for representing descriptions, but not so good when searching for these descriptions. As each dimension is divided by equal increments then it is possible to describe the space as an array with a set number of locations. The number of locations for a 2D array with 20 increments is 400. With a 4D array of 50 increments the number of locations increases markedly to 6 million. There are two ways of overcoming the problems of large search spaces such as this.

- a) Use heuristics to guide the search (see Section 3.5)
- b) Invest knowledge in the data structure (see Section 4.2)

Heuristics are rejected because they do not capture the probabilistic nature of the search space. For example, best first search only measures the space in terms of the distance from the goal, and takes no account of the

interrelations between factors. To overcome this limitation, knowledge has been invested in the space through the application of probability measures on features. Knowledge is characterised in the space through the application of a gravity function. Surrounding each object is a gravity field that pulls inwards, acting as a procedural knowledge function. For example, a small circular gravity field around a object located at (10 10 10) may have values of 2 stored in locations (11 10 10) (9 10 10) (10 11 10) (10 10 9) and values of 1 stored in locations (12 10 10) (8 10 10) (10 12 10) (10 10 8). Some of the fields overlap and are not always even, but depend on the relationships between the features for a specific object. Each feature is tied to the gravity field through its shape and this is important to the expression of the interdependence between dimensions in the database. Those locations which are not occupied by a pointer or a gravity value are left empty.

The database contains links to frames outside the hierarchy, and each of these database frames represent knowledge relevant to the originating feature location. The database is, thereby, a knowledge store of expectancy.

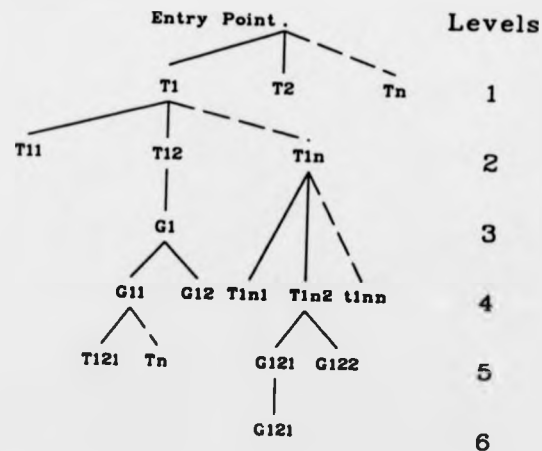
7.3.1.2 Searching for an Object

Each location in the conceptual database can be one of three types: an empty location that has a neutral effect on surrounding locations; a gravity value that dictates strength or pull in a particular direction by an increasing value; and an object address that points to a frame(s).

Object retrieval involves entering the conceptual space at a predetermined location (see next sub-section) and searching for the nearest object through transforming the matrix location. The object is not necessarily the nearest in terms of matrix transformations, but in terms of the closest gravity field and/or of greatest strength. Search, thereby, starts at the entry point, and moves equally in all directions until a gravity field is encountered. The entry point is then moved and search begun again, until a larger gravity value for the entry point is found. Because gravity increases towards an object, search eventually encounters an object address. Set theory is used to restrict the search by moving previous matrix transformations of each dimension (n) to a marked set $M(n)$, adding all possible transformations to an expandable set $E(n)$ by a set transformation step (b), and using the difference in the two sets to create a boundary set $B(n)$ that surrounds the entry point $P(ij..n)$, preventing search from moving within that space. $M(n)$ is increased until a new entry point is found where upon $M(n)$ is initialised with $B(n)$, which itself is initialised with the new entry point. A new $E(n)$ is created, but constrained by $B(n)$.

$M(n) = E(ijk..x) \cup P(n)$	entry points marked
$E(n) = (E(i+1bj+1b..x) \cup P(n)) - B(n)$	entry point(s) expanded
$P(n) = E(n) - M(n)$	new entry points created
$B(n) = M(n)$	boundary created

When the entry point is surrounded by a large area of blank space, search is conducted in a breadth first manner, exploring all the transformations of the surrounding matrix locations. As the search moves outwards $M(n)$ gets larger and larger. By initialising $B(n)$ with $M(n)$, and $M(n)$ indirectly with $E(n)$, $M(n)$ only increases in multiples rather than to the power of the array space. This makes the search acts like a ripple through the search space until it hits a gravity field, where upon the centre of the ripple is relocated to the gravity field. It is at this point that breadth first search is replaced by a reverse hill climbing routine, which homes in on the object at the centre of the gravity field (see Figure 7.5).



T = null transformation
 G = gravity transformation
 O = object transformation

Figure 7.5 The Conceptual Database Search Process.

Magnitude restrictions can be placed on the search by changing the size of the transformation of the original entry point and subsequent locations lying outside the boundary. The larger the restrictions the bigger the steps and consequently the "rougher" the search. Thus:

as $P(i).P(j) \dots P(n) = 1.0$,
then $b = 10$ (maximum step size)

A large restriction can be placed on the search if there is certainty about the search process i.e as the joint probabilities of each matrix $P(n)$ approaches 1.0 Where there is a high degree of uncertainty a low restriction would be placed on search. This certainty depends on meta-level knowledge about the entry point in the search space. A high level of confidence in the location of the entry point gives a high magnitude restriction and a course search. Thus:

```
IF      P-Mat-1 > .7 and
        P-Mat-2 > .7 and
        P-Mat-3 > .7
THEN
        (FDEL 'SETTINGS 'STEP) and
        (FPUT 'SETTINGS 'STEP 'VALUE 3)
```

The reverse is true of an uncertain entry point. Thus:

```
IF      P-Mat-1 < .4 and
        P-Mat-2 < .4 and
        P-Mat-3 < .4
THEN
        (FDEL 'SETTINGS 'STEP) and
        (FPUT 'SETTINGS 'STEP 'VALUE 1)
```

Confidence is a useful measure, because it allows the system to express the certainty of evidence in favour of a hypothesis, and consequently introduces the notion of possibility as well as probability (see Section 4.3.2). It is also interesting to note that of the featured based probability allows uncertainty to be stored in database form, thus a very uncertain point in the database would always lie at the most central part of the space, whereas a certainty point lies to towards the perimeter. This database representation permits both search and knowledge to be expressed in terms of certainty.

7.3.2 The Analogical Reasoner

The aim of the analogical reasoner is to link the conceptual database, to the expert system core through the four stages of target selection, source selection, mapping and evaluation. This forms the outer cycle of accomodation->maturation->affiliation for the overall model. To perform this task the reasoner has to meet the following requirements:

- a) Control the analogical process
- b) Construct a target description, and locatable store.
- c) Select a source description from the conceptual database.
- d) Reconfigure both target and source descriptions for comparison.
- e) Define the mapping process. What is mapped to what.

f) Evaluate the success of the mapping process, and consequently the suitability of the source.

These requirements are addressed in three sections: the control of the analogical cycle, examining the overall analogical cycle; the control sequence, which is the way information is handled between the core, agenda, and analogical reasoner; probability assessments, that drive the selection of analogs; and the stages of analogy, examining in detail each of the four analogical processes.

7.3.2.1 Control of the Analogical Cycle

The complete analogical cycle is based on the continuous analogical reasoning defined by Eskridge (1989), and surrounds the core of the expert system. Tasks are defined in the data dictionary (see Figure 7.6) according to what stage in the analogical cycle they are required.

```
Frame name: (DICTIONARY-RC
(LAB-SET-UP (TARGET WAVELENGTH STEPS SCAN)
              (SOURCE)
              (MAP)
              (VALUATE PEAK-COUNT ASYMMETRY PEAK-HEIGHT))
(SUBSTRATE  (TARGET MATERIAL ORIENTATION)
              (SOURCE HALF-WIDTH)
              (MAP DESCRIPTION)
              (VALUATE PEAK-SHAPE)))
```

Figure 7.6 The Structure of the Deep Data Dictionary

Target tasks are labelled TARGET, source selection tasks SOURCE, mapping tasks MAP, and evaluation functions EVALUATE. Only one analogical process can be active at a time, and communicates with the core via the agenda. As tasks are ascribed to analogical processes, when a particular analogical stage is active, only tasks belonging to that set are placed from the knowledge base of the core onto the agenda. All tasks belonging to other analogical sets are excluded from the agenda. As the stages of analogical reasoning change, so the core tasks are moved through the analogical process. Figure 7.7 illustrates the cycle between the core and the analogical reasoner.

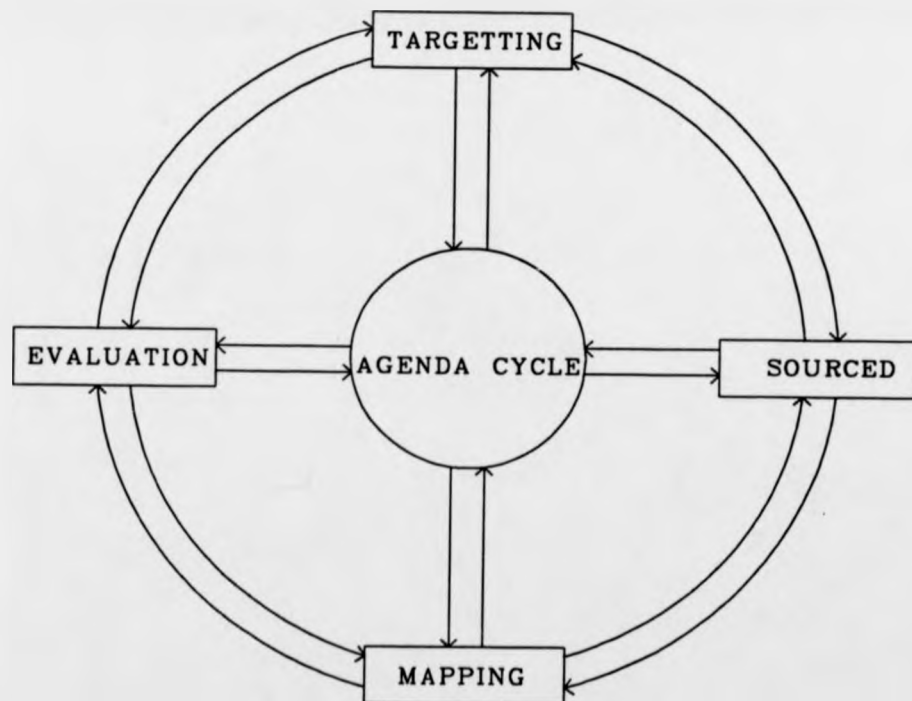


Figure 7.7 The Analogical Cycle of Expert System Shell.

The cycle can move in either direction so that at any time a TARGET description can be built, re-built or modified, many possible SOURCES found, and several MAPPING and EVALUATION processes carried out during a consultation. Movement between each analogical process is controlled by the analogical reasoner, and will happen when:

- a) The necessary information required for the active analogical process is found.
- b) The active analogical process is not extracting any valuable information.
- c) All tasks for the selected analogical process have been exhausted.

For deep reasoning the agenda has been re-defined (see Figure 5.11 for original model). It is made-up of four sections and processing moves from the input section across to storage section via a set of calculations and analog assignments. The agenda receives inputs from the frame system in terms of slots for which values must be found, and assigns priority levels to each slot based on the analog category of slot. The slot assignment is determined by the frame called the dictionary (see Figure 5.12), which classifies slots as being either part of the targeting, source selection, mapping or evaluation process of the analogical reasoner. Control is then passed to the frame system which finds values for each slot and then returns

the results to the agenda. Based on the ratio of success to failure when finding slot values, the priority level of the associated analog and subsequently any assigned slots are adjusted. Because the agenda only passes the top priority slots back to the frame system for filling, slots associated with lower priority analogs will only be filled towards the end of the consultation (see Figure 5.11). Figure 7.8 outlines the processing flow of the agenda.

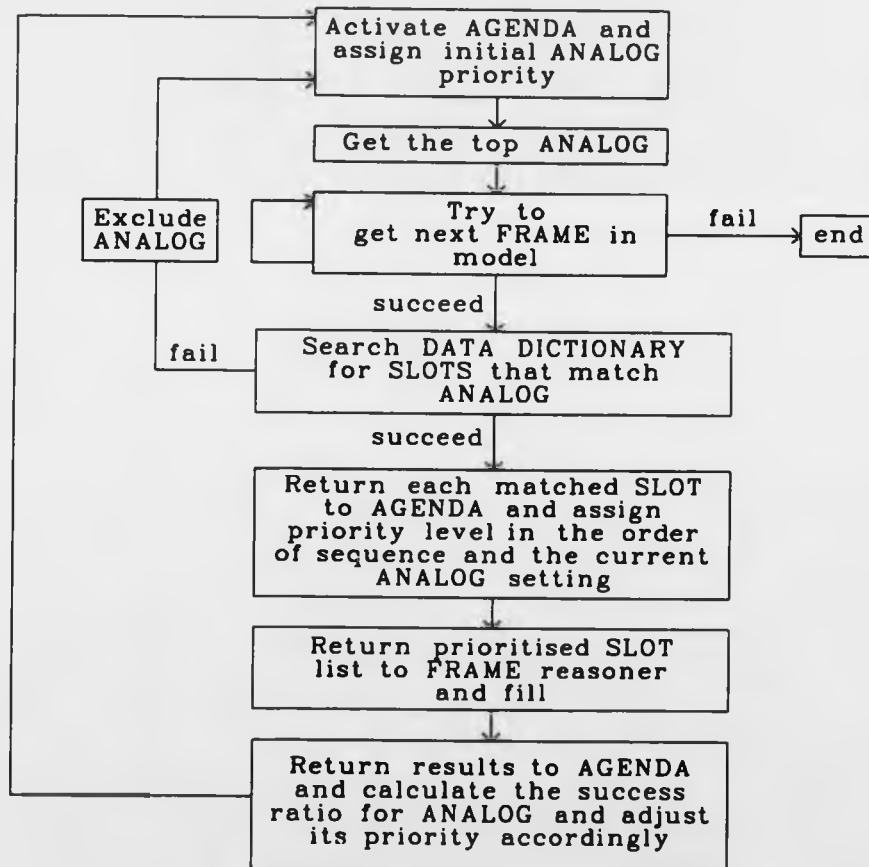


Figure 7.8 The Control Process of the Agenda

7.3.2.2 The Control Sequence for Analogical Reasoning

The agenda, frames system and analogical reasoner form three critical components in the expert system that together search the frame system frame by frame until the slots associated with the top analog are satisfied. In between each frame, the agenda assesses the success of the current top analog and either rewards or punishes it by raising or lowering its priority level, correspondingly raising or lowering the priority levels of the unselected analogs. Thus, when the top analog is successful at filling slots the analogs diverge, and when it is unsuccessful at filling slots the analogs converge. When a new analog has a priority level above that of the previously selected analog the former analog is selected even though all the slots in the previous top analog may not have been satisfied. Generally, therefore, the control process is designed to move between the different stages in analogical reasoning when evidence cannot be found by the other inference engines; frame engine (see Section 5.4.2.1), production rules engine (see Section 5.4.2.2), and the demon logic engine (see Section 5.4.2.3), to support the continued use of this stage in analogical reasoning.

The agenda inspects the current frame for slots that are classified under the top analog. When a list is found control is passed to the frame reasoner which returns an ordered list of active data slots from the control slot of the frame. Control is returned to the agenda where the priority of the top analog and the next analog is compared and the difference between the two divided by the number of

active slots return by the frame reasoner. Each slot is added to the to-be-completed part of the agenda and assigned a sequential decremental values so that the first slot is only decremented once, and second slot decremented twice and so on. The agenda releases the slot with the highest priority to the frame system and passes control once again back to the frame reasoner. The frame system attempts to fill the slot using forward chaining, generating a search tree according to the toggle setting of the system. If a value is found for the slot then it is placed in the Tasks Complete section of the agenda. If no value is found then it is placed in the Task-Incomplete section of the agenda. Once the slot has been placed into one of these two sections it is removed from the to-be-completed section. The next slot from the to-be-completed section of the agenda is then selected by the control system and passed to the frame reasoner for filling, repeating the process. This control cycle of agenda -> frames -> agenda continues until either no more slots exist in the to-be-complete section of the agenda or the analog of the system changes, in which case all remaining to-be-completed tasks have their priority values re-assessed.

7.3.2.3 Probability Assessments of Analogical Reasoning.

At the start of each consultation, labels for all four stages of analogical reasoning are stored as slots on the agenda with an associated priority value. TARGETting=0.4, SOURCE selection=0.3, MAPping=0.2, and eVALUATION=0.1. As

targeting has the top priority it becomes the first active stage in analogical reasoning. Conditional probability is used to assess the success of the active stage in analogical reasoning via the core, hence, the joint probability for all analogs $(0.4 + 0.3 + 0.2 + 0.1)$ always equals 1.0. Tasks associated with top analog are added to the agenda frame by frame, and conditional probability applied to the selected analog frame by frame. Five factors are taken into account when adjusting the analog priorities:

- a) The distance of the task frame from the query frame
- b) The length of the root of Inheritance.
- c) The ratio of filled to unfilled slots for the frame.
- d) The previous priority level of the analog.
- e) The number of analogs that could potentially be active.

To achieve the analogy cycle, all analogs (A) are assigned to either the correct analog set (SET I), or the incorrect analog set (SET II). The active analog (AI) occupies SET I and the others (AII) occupy SET II. In accordance with conditional probability, the probability that the analog is correct is given by $P(AI|AII)$. For all events the conditional probability is therefore:

$$P(AI|AII) = \frac{P(AI) \cdot P(AII|AI)}{\sum_{k=1..n} P(AI) \cdot P(AII|AI)}$$

Where k = number of tasks

Conditional probability is only applied to the active analog $P(AI)$. The probability for other analogs is $P(AII)=1-P(AI)$. The adjustment for each of the other analogs $P(Ai|AII)$ of SET II is proportional to their initial probabilities, thus:

$$P(Ai|AII) = \frac{P(Ai) \cdot P(AII)}{P(AII|AI)}$$

The change in probability of AI is $P(E)=P(AI|AII)-P(AI)$. If $P(E)$ is negative then the two sets converge, and if it is positive they diverge. This degree of convergence and divergence is not linear. Change is greatest when evidence is acquired at a point when both sets have an equal probability ie $P(I)=0.5$ and $P(II)=0.5$, and least when $P(I)$ approaches 1.0. Furthermore, an adjustment factor is included to increase or decrease the amount by which the evidence changes the probability rating of the active analog. The adjustment $P(R|T)$ is based on the length of the root from the top of the frame hierarchy to level of the query (R) combined with the level of the tasks contributing the evidence (Tk):

$$P(R|T) = \frac{\sum_{k=1..n} Tk}{R \cdot n}$$

$P(R|T)$ is conditionally applied to $P(E)$ for the direction and location of evidence towards AI (see Table 7.2).

Table 7.2

The Conditional Application of $P(R|T)$ to $P(E)$

Set Direction	Location of Task in Frame Hierarchy	
	Top Half	Bottom Half
Convergence	$P(R T) \cdot P(E)$	$P(R T) \cdot P(E)^2$
Divergence	$P(R T) \cdot P(E)^2$	$P(R T) \cdot P(E)$

This conditional formula is applied to $P(AI)$ to give an adjusted $P(AI|AII)$ and $P(AI|AII)$ thus:

$$*P(AI|AII) = P(R|T) \cdot P(E) \text{ or } P(R|T) \cdot P(E)^2$$

and

$$*P(AI|AII) = \frac{P(AI) \cdot P(AII)}{*P(AII|AI)}$$

As $P(E)^2$ reduces the effect of evidence on the active set, the active analog is rewarded when evidence is gathered that is specific and close to the query, and punished when evidence is not gathered at the most general level and distant from the query. This heuristic application of evidence, thereby, assumes that tasks at the top of the hierarchy should be known, and that tasks at the bottom are more specialised and less likely to be known. This means the behaviour of the system becomes reward sensitive the closer

the user is to the query, and punishment sensitive the further the user is to from the query. Furthermore, the length of path affects this process. A long path differentiates rewards and punishments. A short path integrates rewards and punishments, and for a path length of one rewards and punishments are equal. This is a logical heuristic since complexity in the frame system is represented by a long root and simplicity by a short root. This heuristic characterises the analogical cycle in two ways. Firstly, if the problem is complex, then the analogical cycle is more sensitive to the evidence collected during inference. This is important in hypothesis formation because there is likely to be less certainty in the evidence and so a greater need to explore options. In operational terms this means more mapping operations. Secondly, with a simple problem the analogs are stable during inference, and mapping may occur only once. This is conducive to the way experts may solve simple problems. Here it is assumed that the reduced number alternatives will be proportional to the reduced number of hypotheses. These jointly constitute a set of meta-rules called success-failure ratio rules.

7.3.3 The Analogical Processes

Each of the four analogical processes operate in series according to the probabilistic analysis of the user knowledge base interaction. The order of operation is initially fixed as Target->Source->Map->Evaluate. However, the cycle varies according to the success-failure ratio

rules. In addition to these there are a further set of meta-rules for pruning the analog selection process. These are called logical-cycle rules. They are simple logical rules that operate as a set of flags within the inference engine of the frame reasoner and state:

Rule 1 WHEN consultation is start
THEN source-flag IS on AND
map-flag IS off AND
evaluate-flag IS off

Rule 2 WHEN source-flag IS on AND
source-tasks IS complete
THEN source-flag IS off AND
map-flag IS on AND

Rule 3 WHEN map-flag IS on AND
map-tasks IS complete
THEN map-flag IS off AND
source-flag IS on AND

Rule 4 WHEN map-flag IS off
THEN evaluate-flag IS on

Rule 5 WHEN evaluate-flag is off OR
map-flag is off
THEN source-flag is on

These rules ensure that mapping only takes place if a source has been calculated or re-calculated, and that no evaluation can take place unless something has been mapped to the target.

7.3.3.1 Target Description

To construct a target description it is necessary to identify elements from the target problem that structurally

map to the target description. It is apparent from both the cognitive view point and that of machine learning (see Section 7.1.2) that large target descriptions need pruning. The core of the expert system is the mechanism for building a description, and to do this it is necessary to integrate the target process into the core. Having built a target description it is then necessary to reduce this to a critical set of descriptions, in this case target features, to allow comparisons between the target and source. STM acts in a similar manner by using bottom-up processing to reduce the size of sensory information to a featural description. Target reduction can be achieved in the core by categorising knowledge into two sets: target and feature knowledge. The inference of the target set determines the feature set which is then stored within the common database of the expert system.

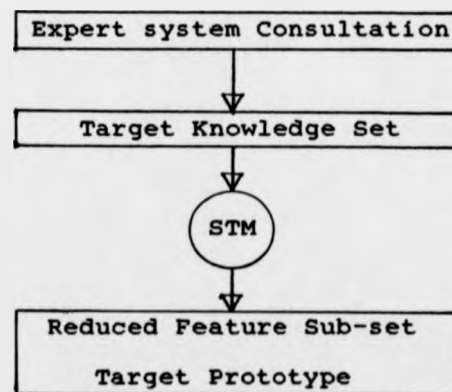


Figure 7.9 The Production of a Target Prototype from the Expert System Core.

The feature set can be seen as the result of STM processing, producing a target prototype from the consultative process of the expert system core (see Figure 7.9).

7.3.3.2 Source Selection

Once a target prototype has been produced from the consultation, a suitable source is selected from the conceptual database. To achieve this an entry point into the probability array of the conceptual database is required. To perform this operation, the reduced feature set is matched to the feature dimensions of the conceptual database. This is like a LTM retrieval process, using analogical reasoning to access a suitable source to match the sensory inputs configured in STM. As probability is the means of describing the source, re-configuring the target prototype into probability dimensions is the most effective way of finding an entry point. This can be achieved using conditional probability on the target prototype. Each feature of the target prototype is matched separately to specified features of the conceptual database, and given a probability rating. The ratings for these target features is elicited from expert. The inference procedure of the core can now be used to propagate Bayesian logic during the consultation (see Section 4.3.2). If the target feature is applicable then the probability rating for each dimension is propagated directly ($P(n)$), but if it is negated then it is taken away from the sum of world probabilities ($1-P(n)$). In conditions where the relationship between the target feature and the conceptual

dimension(s) is unknown, or not specified, no propagation occurs. This means that there is an inexact match between the target and source, which is maintained through the use of conditional probability. When all tasks associated with the target descriptions are complete, the result is a key target description described in probability terms. In other words, an entry point into the conceptual database (see Figure 7.10).

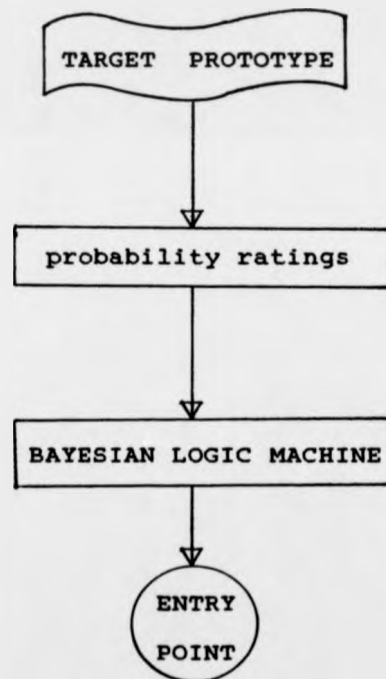


Figure 7.10 The Generation of an Entry Point into the Conceptual Database from a Reduced Target Feature Set.

The entry point is continuously updated whilst the target prototype is being formulated according to the BAYSIAN function:

```
[DEFUN BAYSIAN
  (LAMBDA (PI PITO PTI PITn)
    (SETQ PITn
      (/ (* PI (* PITO PTI))
         (\+ (* PI (* PITO PTI))
              (* (\- 1 PITO) (* (\- 1 PI) (\- 1 PTI))))))
  )])
```

where:

Old probability $P(H E)_o$	= PITO
Base probability $P(H)$	= PI
Probability of new evidence $P(E)$	= PTI
Probability given new hypothesis $P(H E)_n$	= PITn

The application of the Bayes's rule to the target prototype configures the large description down to the reduced source location using the elicited key features (see Section 6.5.2). In doing so, the combinatorial explosion that would occur if matching on all features is avoided. Once this location is set, the search for the source prototype begins using the conceptual database algorithm (see Figure 7.5). The search process and the consequent retrieval of the schema from LTM is controlled by the source selection procedure of the analogical reasoner. Source selection tasks, which reside within the core, are selected from the data dictionary and propagate into the structure reflecting of the procedures associated with the corresponding slots in

the frame hierarchy. All source tasks that contribute to the source location are stored in feature tables with assigned probability values elicited from the expert (see Figure 7.11). Each feature table represents one dimension of the conceptual database and has a label describing the feature. There is also a Bayesian slot that represents the probability value associated with the dimension. Following these standard slots are all the associated source slots that model the dimension.

Frame: Dimension M1

Label	: Complex-Peak-Type
Bayesian	: 0.5
Satellites	: 0.8
Grading	: 0.7
Few-layers	: 0.3

Frame: Dimension M2

Label	: High-Peak-Density
Bayesian	: 0.5
Satellites	: 0.3
Near-perfect	: 0.7

Frame: Dimension M3

Label	: High-Peak-Count
Bayesian	: 0.5
Thin-layers	: 0.7
Cap	: 0.6

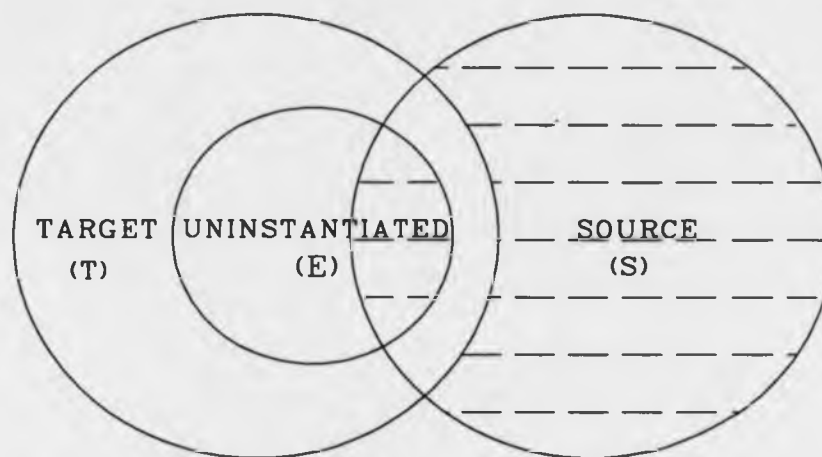
Figure 7.11 Feature Tables for X-ray Rocking Curve Domain

To start of a consultation all dimensions are central. Each feature associated with a dimension is equally likely to be

true or false. The entry point is, therefore, at the centre of the conceptual space, occupying the most uncertain position. When source procedures are active, every request from the agenda activates the Bayesian logic engine. This checks the feature tables to see if any of the current tasks can be associated with the conceptual dimensions. Any that do, are used to propagate the Bayesian value associated with the frame. This increases the certainty of the source description, and moves the entry point. When the source selection analog completes its processing, the entry point to the conceptual database is configured into an array location, and the search algorithm activated. The search process returns the pointer of the object that is conceptually the closest to the entry point in the database. The pointer either points to a single object or an object hierarchy in a stack of frame hierarchies stored in LTM behind the consultative frame.

7.3.3.3 Mapping

Mapping is the process of adding the new data located in the source to the target. Within the frame system this involves adding slots from the source to the target. The query frame and all frames above it in the target hierarchy are matched by frame name to the source frame(s). Any new source slots or new source procedures or defaults are added to the target frame provided they have not already been satisfied on the agenda. A Venn diagram can be used to express the mapping process (see Figure 7.12).



Shaded Area = slots MAPPED from the SOURCE to the TARGET.

Figure 7.12 Venn Diagram of the Mapping Process

Slots added to a matched target and source frame can be expressed thus:

$$(FNS)U(S-(TNS))$$

When the mapping procedure is complete, any mapping tasks that have been defined by in the data dictionary are placed on the agenda. This time, however, additional mapped knowledge is available to the core of the expert system. This has the effect of focusing the reasoning on the mapped

tasks, thereby, equating the prototype target description to the source prototype description.

7.3.3.4 Evaluation

Evaluation is the last analogical process of the reasoner. This assesses the success of the mapping process by comparing the 'information gain' achieved by mapping the slots. Information gain is defined as the difference in the probability of filling a slot before and after mapping. If there is a trend towards instantiation then the mapping process is adjudged to have been successful. To achieve this requirement three factors are taken into account:

- a) The distance of each task from the root query. The closer the task is in its origins to the frame from which the consultation was initiated the more important it is considered.
- b) The matching of the values in the source frame(s) to the target values following the evaluation cycle.
- c) The amount of knowledge obtained in the target as a result of the application of source to the target.

Taking in to account these factors, it is possible to classify the interactions between the target and source using set theory, where μ = all the slots involved in the consultation, each element of a set is in lower case (a b c

d), A = target set, C = source set, B = target slots with values, D = source slots with values, and B subset A , and D subset C .

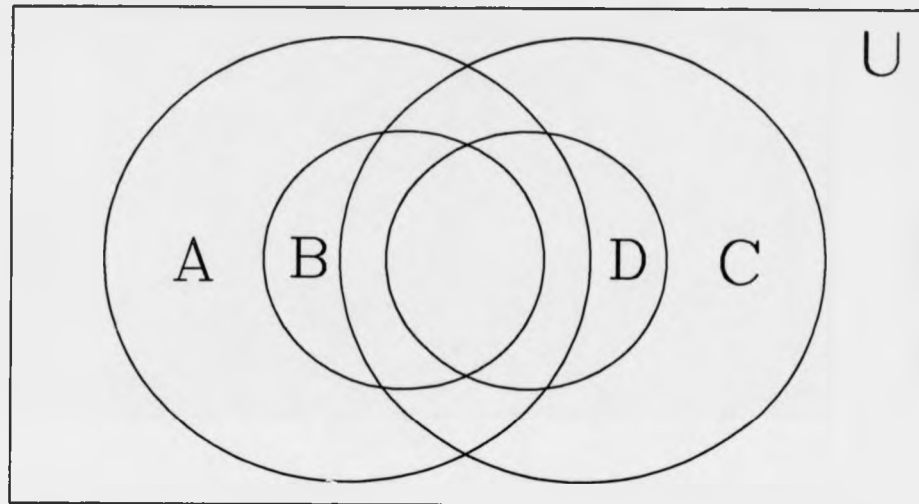


Figure 7.13 Set Definition for Evaluation Function.

From the set definition, five important sub-sets can be derived for a single target and source frame.

(AUC)	{all tasks}
(ANC)	{shared tasks}
$(\exists b=d \ B \cap D)$	{shared tasks with shared values}
$(B \cap D)$	{shared tasks with any values}
$(B \cup D)$	{all tasks with values}
$((AUC) - (B \cup D))$	{tasks with no values}

Provided that:

$$(ANC \text{ neg } \emptyset) \text{ or } (BND \text{ neg } \emptyset)$$

Each of these sub-set can now be used to describe the analogical reasoning task in terms of probability. The probability of achieving a matched task $P(TES)$ is:

$$P(\{b=d \mid BND\}) / P(ANC).$$

The confidence in the matching process $P(T^{\wedge}S)$ is:

$$P(\mu) - (P(AUC) - P(BUD) / P(AUC)).$$

The similarity between the source and target $P(T \leq S)$ is:

$$PANC / (P(ANC) \cup P(BUD)).$$

The success of the match for a frame is:

$$P(TES) \cdot P(T^{\wedge}S) \cdot P(T \leq S) / \mu.$$

Each set of equations can now be combined for every frame ($F_1 \dots F_n$) to give an evaluation of the match for all frames for which mapping occurs. This combined mapping of target and source frames is isomorphic, but, as mentioned earlier, this is not the case for slots because both their whole structure and parts of it can be attached or appended to the target frame from the source frame.

Given the target frames (Tn) of the consultation, and the source frames (Sn) from the database, the evaluation of the mapping process for all frames operating by propagating bayesian logic is:

$$E_k=2..n \frac{P(T_{k-1} \# S_{k-1}) \cdot P(T_k \# S_k)}{P(T_{k-1} \# S_{k-1}) \cdot P(T_k \# S_k) + (1 - P(T_{k-1} \# S_{k-1})) \cdot (1 - P(T_k \# S_k))}$$

$$E_k=2..n \frac{P(T_{k-1} \wedge S_{k-1}) \cdot P(T_k \wedge S_k)}{P(T_{k-1} \wedge S_{k-1}) \cdot P(T_k \wedge S_k) + (1 - P(T_{k-1} \wedge S_{k-1})) \cdot (1 - P(T_k \wedge S_k))}$$

$$E_k=2..n \frac{P(T_{k-1} \triangleleft S_{k-1}) \cdot P(T_k \triangleleft S_k)}{P(T_{k-1} \triangleleft S_{k-1}) \cdot P(T_k \triangleleft S_k) + (1 - P(T_{k-1} \triangleleft S_{k-1})) \cdot (1 - P(T_k \triangleleft S_k))}$$

Each measure of analogy can be combined to give the overall evaluation of all frames involved in the mapping process:

$$P(T \# S) \cdot P(T \wedge S) \cdot P(T \triangleleft S) / \mu$$

Make # etc bigger to indicate that all frames not just one.

The overall evaluation is compared with the probability assessment generated by the target analog, and if it is higher in value then the evaluation is considered successful and the consultation continues with the source selection tasks eliminated, otherwise they are included for further matching later in the consultation.

7.3.3.5 Consolidation

Consolidation is not part of the continuous analogical reasoning cycle, but the result of long term evaluation of the analogical process. It is the mechanism by which the outer cycle of the deep expert system model operates and the learning model by which the expert system dynamically alters the conceptual database. Consolidation takes place when the mapping of source to target produces a high level of confidence $P(T^{\wedge}S)$ in a very similar source and target $P(T \leftarrow S)$, but with a low level of matching ($P(T \equiv S)$). The probability that consolidation $P(V)$ will take place with a defined degree of change $P(CH)$ is dependant on a constant relationship (see Table 7.3).

Table 7.3

Probability of Consolidation given the Evaluation
of Target and Source.

$P(T \equiv S)$	$P(T^{\wedge}S)$	$P(T \leftarrow S)$	$P(V)$	$P(CH)$
High	High	High	Med	Low
High	High	Low	Low	Med
High	Low	High	Low	Low
High	Low	Low	Low	Med
Low	High	High	High	Med
Low	High	Low	Med	High
Low	Low	High	Med	Med
Low	Low	Low	Low	High

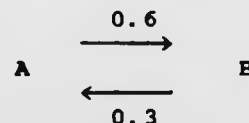
The following equation is used to determine the probability of consolidation given $P(T \equiv S)$, $P(T^{\wedge}S)$, and $P(T \leftarrow S)$:

$$P(W) = \frac{P(T^{\wedge}S) \cdot P(T \leq S) \cdot P(T \leq S)}{P(T^{\wedge}S) \cdot P(T \leq S) \cdot P(T \leq S) + 1 - P(T^{\wedge}S) \cdot 1 - P(T \leq S) \cdot 1 + P(T \leq S)}$$

Having established the probability of consolidation, three possible changes can be effected corresponding to the long term changes of accommodation, maturation, and affiliation. These changes affects both LTM (the configuration of frames behind the current consultation) and STM (the configuration of the conceptual database) of the outer consultative cycle.

7.3.3.5.1 Accommodation

The process of accommodation is defined as the re-configuration of frames in the frame stack. When the connections within the stack are changed, the links from STM to LTM change, and object locations in the concept space point to different objects configurations in the frame stack. All frames in LTM have the same hierarchy, but not the same slot configurations, values or procedures. The net is an arrangement of interconnections between slots and there associated values and procedures. These interconnections are weighted as a function of the strength, 1 is very strong, 0 is no connection. The weighting has two directions of pull, thus:



If a slot achieves a value of 1 then it moves to the linked frame, if the slot achieves a weighting of 0 then the connection between slots is broken. All slots are initially unconnected, but as the consultation process progresses it forges and breaks links as a results of weighting adjustments. The weighting in itself does not affect the transfer of data between the LTM network, but operates as a all or nothing device, that stops once the link is broken and starts once a link is made. A set of rules are defined to control the transfers of slot values and procedures, and are as follows:

- a) Links are formed between the selected slots of the source and slots in the adjacent frames of LTM.
- b) A link is only formed with a slot from another frame if the current slot in the selected source does not match the value in the consultation.
- c) When a link is formed it is strengthened if the values in the selected source do not match the consultation value and the linked slot matches this value.
- d) A link is weakened if the selected source values matches the consultation value, and the linked slot does not.
- e) If neither the selected source value(s) do not match the consultation and the adjacent frames in the stack do not,

then a chained search is conducted outwards until values are found that do match.

f) As the chaining moves outwards from the source, the propagate strength increases. This accelerates distant value and procedures towards the source location.

g) When a value is move towards a source, the link is broken between the origin of the value and the location to which it moved.

h) When a value is moves towards a source it only moves one frame stack at a time.

In applying these rules of accommodation, LTM dynamically changes as a result of consultative behaviour. These changes are governed by the Baysian logic engine, using the same constraints as applied to the evaluation function.

7.3.3.5.2 Maturation

Maturation is defined as the re-ordering of the concept space between key features of STM. This is effectively the moving of object locations in the conceptual space, along with their gravity fields. To move a gravity field, the entry point at the start of the search is taken from the mapping location at the end of the search. For example, given the search tree:

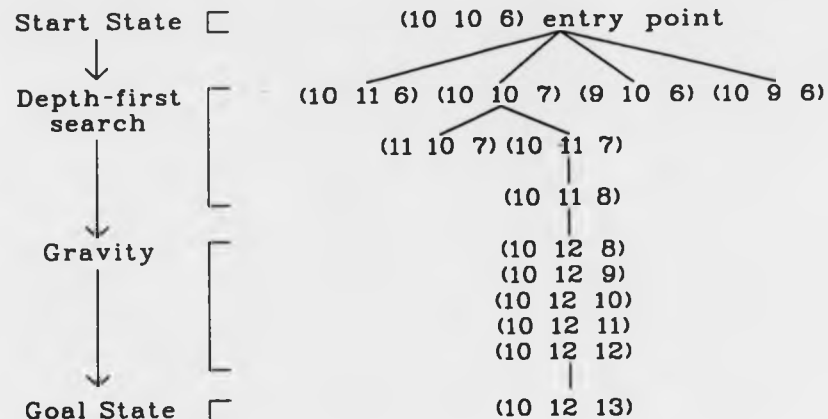


Figure 7.14 Search Tree of Conceptual Database

the probability shift $P(n)$ is the difference between the start state (s_n) and the goal state (g_n) multiplied by the probability increment $P(i)$ for each search step i e:

$$[P_i \ P_i \ P_i] \cdot [s_1 \ s_2 \ \dots \ s_n] - [g_1 \ g_2 \ \dots \ g_n] = [p_1 \ p_2 \ p_3]$$

$$[0.05 \ 0.05 \ 0.05] \cdot [10 \ 10 \ 6] - [10 \ 12 \ 13] = [0 \ -0.1 \ -0.35]$$

The strength of the analogy is used to move the goal state toward the start state constrained by the conditions of consolidation (see Table 7.3). Conditional probability is applied to the probability shift to change the goal location. If perfect analogy is achieved the goal state moves to the start state. However, in practice this is never achieved, and, therefore, movements are more limited. The

strength and direction of movement has two contributing factors:

- a) The Overall strength of the analogy .P(Ψ)
- b) The probability of movement P(d) of each dimension or key feature (k) away from its central location (0.5) in the Bayesian table.

By conditionally applying each of these factors, a analogous movement to the source is achieved thus:

$$\frac{P(n)k.P(\Psi).(P(d)k-0.5)}{P(n)k.P(\Psi).(P(d)k-0.5)+(1-P(n)k).(1-P(\Psi)).(1-P(d)k-0.5)}$$

If the overall probability of analogy is 0.8, the probability shift [0 -0.1 -0.35], and the baysian values of each feature [0.3 0.8 0.7], then the accommodation calculation is as follows:

$$\begin{aligned} & \frac{[0].(0.8).[0.-2]}{[0].(0.8).[0.-2] + 1-[0].1-(0.8).1-[0.-2]} & = 0 \\ & \frac{[-0.1].(0.8).[0.3]}{[-0.1].(0.8).[0.3] + 1-[-0.1].1-(0.8).1-[-0.3]} & = -0.18 \\ & \frac{[-0.35].(0.8).[0.2]}{[-0.35].(0.8).[0.2] + 1-[-0.35].1-(0.8).1-[-0.2]} & = -0.35 \end{aligned}$$

The maturation shift is a proportional shift of the maximum possible, giving a shift:

[1 -1 -1].[0 -0.1 -0.35].[0 -.18 -.35].[0.05 0.05 0.05]

of the source from its original position resulting in a new source location of [10 12 11].

The proportional shift of the source is directly transferred to all elements of the associated gravity field, any overlapping values are resolved by the strength of the field rule for each specified location. The movement constitutes a change in STM and one that dynamically reflects the matching between source and target. Movement trends can be stored to stabilise the source location, with radical shifts on location having an increasingly minimal effect on the source as the number of consultation increases. In this sense, the source converges on a best location for the range of problems encountered by the expert system for that particular analogy, and only moves location if there is a persistent movement in one direction that is not covered by another source, such as would occur if the nature of the domain changed.

7.3.3.5.3 Affiliation

Affiliation occurs when the gravity surrounding each object is altered to give a differing relationship between key features in STM. The process changes the shape of the gravity field to reflect this changing relationship. To

achieve a consistent configuration it is necessary to apply constraints for the range of conditions covering the key elements (Dn) of the domain. A set of relationships or a model can be built between all features and expressed by a set of formulae (Fn):

$$\begin{array}{lcl}
 & D1 \cdot 2F & \\
 F1 & = \frac{\quad}{D2} & (Dn) = \text{variables} \\
 & & (Fn) = \text{formula} \\
 & & \text{others} = \text{constants} \\
 F2 & = D2 \cdot L & \\
 F3 & = D4 \cdot D1 & \\
 F4 & = A2 \cdot D2 & \\
 F5 & = D1 \cdot M &
 \end{array}$$

Initially, all key elements are assumed to be independent of each other. However, through the inferencing process, relationships are built between combinations of features, and meta-rules used to define the dependency formulae (Fn). In the example above, the model includes five dependency relationships between key features. The model can be represented as a constraint graph, showing the direction of constraints between the constants and the variables, with the arcs showing the direction of application of each formula (see Figure 7.15). The constraint graph shows five key features, two independent (D5 D3), and three dependant (D1 D4 D2). Constraint evaluation is used to establish the best fit for the dependencies, giving an applicable set of equations for the dependant features.

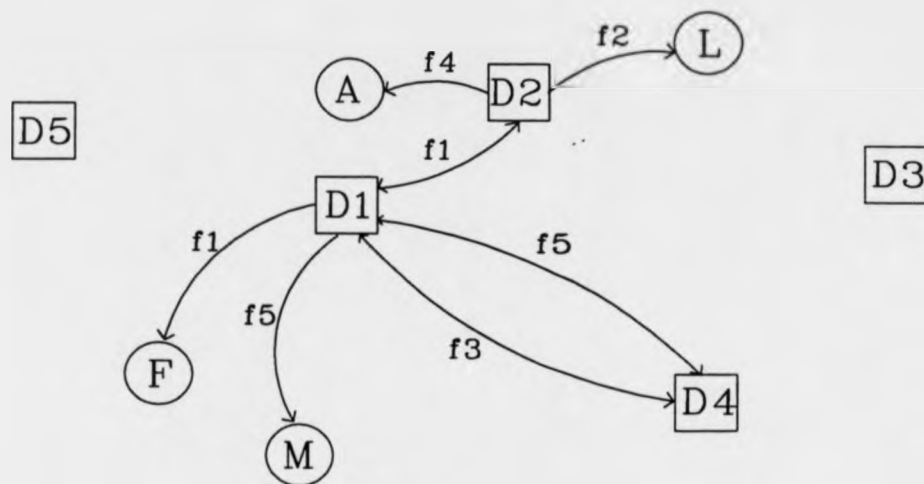


Figure 7.15 Constraint Graph for Affiliation Process

This results in a balanced constraint relationship for the model ie:

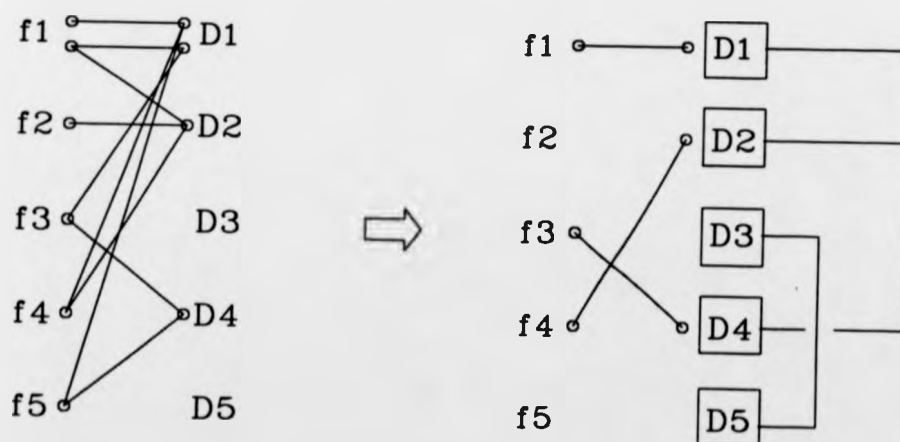


Figure 7.16 Balance Constraint Model for Affiliation

The evaluation gives an applicable set of rules that can be applied to the dependant features. The equations can be integrated with respect to each dimension to give an affiliated boundary based on meta-rule application. This changes the shape of the boundary for the gravity field and, hence, the degree of pull towards the source object. Any dimensions that remain independent of the constraint are not changed.

By altering the shape of the gravity field, the consultation is adjusting STM. The boundary may extend in certain direction, shrink to the centre a selected points, and so on. This changes the probability that a particular source will be chosen as a function of a trend towards the application of consistent constraints. As with maturation, the effects of affiliation reduce as a consultative trend emerges, settling on a prescribed shape to the gravity field.

7.4 The Deep Expert System Shell and the Model

The elements of the expert system shell have been incorporated into the model (see Table 7.1). The core forms the inter-cycle of the model. The deep reasoning system suuounds the core in the same way as the outer learning cycle surrounds the model. The deep elements of a conceptual database, stack of frames, and analogical reasoner form the components of the outer cycle, and the processes of accomodation, maturation and affiliation are the linking processes between these components. There are connections

between each cycle, that corresponds to the links that occur between an individual consultation and a series of consultations over time. This is the three way dynamic relationship that modifies the expert system structure to fit the world in which it operates. The stack of frames connects to the consultation frame, or current schema, through the mapping of source to target, and the consequential evaluation of strength of analogy. The hypothesis generation process links to the algorithmic reasoning mechanisms through the generation of an entry point into the conceptual database, and the result of the search process that is governed by the source locations and gravity fields. Finally, the actual world and user is connected via the modelling of the expertise, and the selection of new analogical tasks as a result of 'information gain' (see Figure 7.3).

7.5 Overview of the X-ray Rocking Knowledge and Deep Reasoning

X-ray rocking curve knowledge has been represented in the core of the expert system using rules, frames, and demons. This knowledge is acted upon when the user places a query on to the agenda. The usual methods of decomposition, inference, and constraint application are used to control the consultation until such a point when no more tasks are left unsatisfied on the agenda.

The key features extracted from the experts using experimental concept formation techniques (see Section 6.3)

are used to configure representations of previous consultations into the conceptual database framework described earlier (see Section 7.3.1). The key features are prototype linker to the internal schemas stored in LTM.

The conceptual database is composed of three dimensions (20 x 20 x 20) for peak count, peak density and rocking curve type. At locations (0 0 0) and (20 20 20) the probability of a high peak count, of a high peak density, of a complex rocking curve are 0 and 1 respectively. All locations within those bounds have probabilities between 0 and 1. Of the 8000 possible locations only 0.5% have pointers. The conceptual database exists along side an expert system core, and is linked to the frame hierarchy via the analogical reasoner. X-ray rocking curve knowledge is classified within the frame hierarchy into one of the four stages of analogical reasoning. All target knowledge consists of:

- a) basic experimental conditions.
- b) expected structure of the sample.
- c) expected rocking curve profile.

At this stage in reasoning a target profile is built of the problem. All the knowledge is registered in triple form within the common database (see Section 5.7). When target knowledge is complete, the source classification tasks are activated. The job of this stage is to describe the experimental rocking curve profile in explicit terms. The description of the rocking curve profile is tied to each of the key features (peak-type peak-density peak-count), and

production rules used to match the target profile to the experimental data. Depending on the outcome of this stage, simulation may be recommended. Once sufficient source data have been added, and provided there is no more high priority targeting, the conceptual database is consulted. The probable source is searched for and when located, mapped to the frame hierarchy. Once mapped, the analog reasoner executes all existing and newly added tasks, classified under the mapping stage. This is the interactive stage of reasoning, recommending changes to the target profile to match the experimental data. Some of these tasks are executed as map tasks, however, other are assigned to the other analogs in the frame hierarchy in accordance with their role in the previous source consultation. Evaluation follows mapping, and aims at comparing the success of the mapping process. During evaluation a successful source is one that increases the probable uptake of knowledge within the system. In terms of X-ray rocking curve analysis, this is the source that confirms the initial hypothesis built in the target profile and composed through the source tasks. If evaluation proves successful, then reasoning stops. If not the cycle continues, finding a new hypothesis and mapping new slots to the frame hierarchy. This interactive cycle continues until:

- a) The target profile is proved correct.
- b) No sources can be found in the conceptual database.
- c) All tasks on the agenda are exhausted.

From the analysis, therefore, it is possible to see that the two staged process used by domain experts is represented as initially, the formation of a target profile followed by a source hypothesis, and iteratively by the continued mapping of conceptual data to match the initial hypothesis.

7.6 Conclusions

A model has been proposed for the development of a deep expert system shell using analogical reasoning. This has been implemented for the domain of X-ray rocking curve analysis. The system performs under a variety of conditions in accordance with what would be expected from a specialised shell. The model, upon which the deep system is based, satisfies the specified protocol with the exception of the learning component. However, a learning framework is proposed using the conceptual database and frame network as dynamic entities. It can be said that the deep model accepts the role of statistical inference within the symbolic framework, forming interconnections between the knowledge systems operating in the expert system shell.

Chapter 8

Conclusions

8.0 Overview

I began with an overview which focused on my interests and intentions in examining deep knowledge within an expert system framework and here I should like to conclude with a brief summary of the approach taken, my findings and the implications of this study.

The importance of a formal design procedure

In designing an expert system it is necessary to initially capture existing tools within the chosen architecture in a manner that is cognitively viable to the domain (see Table 5.3). The outcome of this process is a design architecture from which a system can be built (see Section 5.4). The expert system core for X-ray rocking curve analysis reflects this importance. The core has an range of representations (see Section 5.4.1) and a domain independent inference system (see Section 5.4.2). The core can solve X-ray rocking curve problems, and has an architecture partially suited to the domain. However, the core only represents knowledge in a declarative and conditional manner. The domain of X-ray rocking curve analysis is open ended and visual in nature. Not all the knowledge of the expert is captured in this format. Furthermore, if the same data are presented to the system the same results are achieved. This is consistent, but not very expert. Expert systems for X-ray rocking curve

analysis have to be adaptive because it would be impossible to encode all the possible configurations of such problems within a knowledge base.

8.1 Overcoming the bottle-neck of knowledge based systems

Conventional knowledge elicitation techniques apply to the X-ray rocking curve domain (see Section 6.1). The conclusions of this research are that they are suitable for shallow expert system development, but do not directly extract deep knowledge, in this case iconic representations of X-ray rocking curve data, from the domain. This thesis proposes a new knowledge elicitation technique for deep knowledge extraction based on the research of Posner and Keele on concept formation (see Section 6.3). The results of this elicitation statistically implies that the three key features of Peak Count, Peak Density and Peak Type are used by experts when organising their visual encoding of X-ray spectra (see Section 6.5). These research data are considered an important piece of deep knowledge about the domain, and knowledge that is not captured by the expert system core.

8.2 A search for deep knowledge representation

The core architecture of the expert system is re-analysed in response to deep knowledge needs (see Section 7.2.1). From this, a new architecture emerges based on the principles of continuous analogical reasoning (see Figure 7.2). Deep

knowledge is now defined as the selection and matching of a current problem definition to a previous problem definitions using an analogical framework (see Section 7.3). This is not a case based reasoning approach to problem solving, but an adaptive approach. The previous problem definitions are modified as a result of the matching process between the source and target (see Table 7.1). Depending on the scope of the 'world domain' in which the expert system operates, the consultation with the user is modified. Modification is founded on statistical rating of large descriptions of problems to key features (Peak Count, Peak Density and Peak Type). This technique is a conceptual top down approach to knowledge extraction. It is quite the reverse of the neural net approach, which is a bottom up technique of mapping data to internal configurations.

8.3 Results of an Analogical Framework

Results of using analogical reasoning are that problem solving sessions can be formulated in a rapid manner. The structure of the consultation with the user is generally governed by the previous consultation, and consequently contains the best problem solving sequences for a problem of that type. In using this architecture the guidance for an identified problem (a problem with a defined target description) will follow the principles of analogical reasoning. Given the exhaustion of a target description, new data are mapped from source to target for further instantiation until a point is reached when it is

statistically viable to evaluate the target and source. Consolidation is the result of the post processing of evaluations. This is the adaptive phase of the consultation. Currently, there are no field results of the adaptive nature of the expert system. In principle, however, the architecture has a valid foundation because it poses problems in a formal manner (see Section 7.3).

8.4 Implications of Analogical Frameworks

Because analogical frameworks are adaptive they support architectural features that enable systems that are built from them to encode dynamic characteristics. These characteristics are used within this thesis to advance the field of expert systems, resulting in both deep knowledge encodement and deep reasoning within the domain of X-ray rocking curve analysis. In other words, this research captures not just the declarative and conditional knowledge of a conventional core, but also case based knowledge and adaptive strategy. The expert system for X-ray rocking curve analysis is a second generation system, with a model based reasoning strategy (see Section 5.3.6). The environment within which the architecture is built has the potential for integrating user modelling techniques to redefine the consultation based on the level of expertise of the user. The expert system is modular and could be adapted to work in other domains provided key features are extracted and then rated within the expert system shell. It is not necessary to use the analogical reasoner, and because this principle is based on task allocation (see Figure 7.6) all slots within a

frame can be described as Target Tasks hence omitting the other stages of analogical reasoning. In this case, the expert system will operate as a shallow representation typical of most commercial expert system shells.

8.5 Limitations and Future Developments

Analogical reasoning is limited by the capacity of knowledge elicitation techniques to extract key features from a domain. In terms of search and mapping it is only possible to currently achieve results if large problem descriptions can be reduced to a smaller comparative set of descriptions. This is a limit upon the wide use of analogically based reasoning within expert systems. Furthermore, the accuracy of the source selection is only a function of the accuracy of the statistical rating applied to knowledge in the domain. The statistical basis of source selection will always be a limitation. Logic based analogical systems may provide a solution, but at the expense of a problem reduction prior to source selection. The exploration of this issue may prove important for the advancement of knowledge based techniques.

The adaptive capacity of a system is also questionable. If 'bad consultations' drive the expert system shell, then the source data will eventually adapt in a 'bad manner'. Modification of the conceptual database (see Section 7.3.1) does not necessarily mean an improvement in the expert system. Only expert use of the system will result in a expert conceptual database. To resolve this issue, it is

necessary only to adapt the database if the system is being used by an expert. Here, user modelling, and its capacity to identify expertise, could be of importance.

Finally, the expert system shell needs to be part of a larger knowledge based environment in which all the necessary tools exist to create syntactically correct input files for the expert system and provide debugging facilities. In addition, number of compilers are required to configure the data structures into an efficient form for pattern matching.

Given these recommendations, it is believed that the development of adaptive knowledge based techniques could provide important advances in the capturing of expertise within software architectures.

Bibliography

- Bell, J.L. and Machover, M. (1977). A Course in Mathematical Logic. North-Holland: Amsterdam.
- Bobrow, D.G. and Collins, A. (1975). Representation and understanding. Studies in Cognitive Science. New York: McGraw-Hill.
- Boden, M. (1977). Artificial Intelligence and Natural Man. Harvester Press: Hassocks.
- Chang, C.L. and Lee, R.C.T. (1973). Symbolic Logic and Mechanical Theorem Proving. Academic Press: New York.
- Cullity, B. (1978) Elements of X-ray Diffraction Philippines: Addison-Wesley.
- Davis, R. and Lenat, D.B. (1982). Knowledge-Based Systems in Artificial Intelligence. New York: McGraw-Hill.
- Fillmore, C. (1968). The case for case. In Universals in Linguistic Theory. Bach, E. and Harms, R.T. (eds). Holt: New York.
- Fisher, R.A. (1954). Statistical Methods for Research Workers. (12th ed.). London: Oliver & Boyd.
- Forsyth, R. (1984). Expert Systems: Principles and Case Studies. Chapman Hall: London.
- Hart, A. (1986). Knowledge Acquisition for Expert Systems. Kogan Page.
- Hayes-Roth, F., Waterman, D. and Lenat, D. (1983). Building Expert Systems. London: Addison Wesley Publishing Company.
- Kelly, G.A. (1955). The Psychology of Personal Constructs. Norton: New York.
- Klatzky, R.L. (1975). Human Memory: Structures and Processes. San Francisco: W.H. Freeman.
- Kleene, S.C. (1977). Mathematical Logic. Wiley: Chichester.
- Kowalski, R. (1979). Logic for Problem Solving. North-Holland: New York.
- Lindsay, P.H., and Norman, D.A. (1972). Human Information Processing. New York.
- Marr, D. (1982). Vision. Freeman Press: San Francisco.
- Plutchik, R. (1974). Foundations of Experimental Research. (2nd ed). New York: S Harper & Row.

Rich, R. (1983) Artificial Intelligence. New York: McGraw-Hill.

Ullman, J. D. (1982). Principles of database systems. MD: Computer Science Press.

Winston, P. H. (1975). The Psychology of Computer Vision. New York: McGraw-Hill.

Rich, R. (1983) Artificial Intelligence. New York: McGraw-Hill.

Ullman, J. D. (1982). Principles of database systems. MD: Computer Science Press.

Winston, P. H. (1975). The Psychology of Computer Vision. New York: McGraw-Hill.

References

- Adlassnig, K.P. and Kolarz, G. (1982). Cadiag-2: Computer-assisted medical diagnosis using fuzzy subsets. Approximate Reasoning in Decision Analysis. M.M.Gupta and E.Sanchez. (eds). pp.269-275. Elsevier North Holland: New York.
- Agogino, A.M. and Rege, A. (1987). IDES: Influence diagram-based expert system. Mathematical Modelling. vol 8: pp.227-233.
- Aikins, J.S. (1983) Prototypical knowledge for expert systems. Artificial Intelligence. vol 20: pp.163-210.
- Balzer, R., Erman, L.D., London, P.E., and Williams, C. (1980). HEARSAY-III: A domain-independant framework for expert systems. Proceedings AAAI. vol 1.
- Berliner, H.J. (1973). Some necessary conditions for a master chess program. IJCAI-3. pp.77-85
- Benson, D.B. Hilditch, B.R. and Starkey, J.D. (1979). Tree analysis techniques in Tsumego. Proc. IJCAI-6.
- Boden, M. (1977). Artificial Intelligence and Natural Man. p.346. Harvester Press: Hassocks.
- Boden, M. (1977). Artificial Intelligence and Natural Man. pp.305-306. Harvester Press: Hassocks.
- Boose, J.H. (1986). ETS: A system for the transfer of human expertise. Knowledge Based Problem Solving Kowalik, J.S., (ed). pp.68-111. Englewood Cliffs, NJ: Prentice Hall.
- Bradshaw, J.M. and Boose, J.H. (1987). Decision analytic techniques for knowledge acquisition: combining situation and preference models using AQUINAS'. Special issue on the 2nd Knowledge Acquisition for Knowledge-Based Systems Workshop.
- Breen, T.J. & Schvaneveldt, R.W. (1986). Classification of empirically derived prototypes as a function of category experience. Memory and Cognition. vol 14: pp.313-320.
- Bruce, B.C. (1975). Case systems for natural language. Artificial Intelligence. vol 6: pp.327-360.
- Buchanan, B.G and Feigenbaum, E.A (1978). DENDRAL and Meta-DENDRAL: the applications dimension. Artificial Intelligence. vol 11: pp.5-24.
- Buchanan, B.G. (1982). Mechanising the search for explanatory hypotheses. Philosophy of Science Association. vol 2.

Buchanan, B.G., and Shortliffe, E.H. (1984). Rule-based expert systems. The MYCIN experiments of the Stanford Heuristic Programming Project. Reading: Massachusetts. Addison-Wesley. pp. 45-50.

Buchanan, B.G. and Shortliffe, E.H. (1984). Rule-based expert systems. The MYCIN experiments of the Stanford Heuristic Programming Project. Reading: Massachusetts. Addison-Wesley. pp. 60-61.

Buchanan, B.G., and Shortliffe, E.H. (1984). Rule-based expert systems. The MYCIN experiments of the Stanford Heuristic Programming Project. Reading, Massachusetts. p. 60. Addison-Wesley.

Burton, R. and Brown, J.S. (1979). An investigation of computer coaching for informal learning activities. International Journal of Man-Machine Studies. vol 11: pp. 5-24.

Cheeseman, P. (1985). In defense of probability. Proceeding 9th International Joint Conference on Artificial Intelligence. Morgan Kaufmann. Palo Alto, California. pp. 1002-1009.

Christine, C. and Izak, B. (1991). Case research on knowledge acquisition: observations and lessons. Knowledge Engineering Review. vol 6: No. 2: pp. 97-120.

Clancey, W.J. and Letsinger, R. (1981). Neomycin: Reconfiguring a rule-based expert system for application to teaching. Proc. IJCAI-81. pp. 829-836.

Clancey, W.J. (1984). Classification problem solving. Proceeding of the National Conference on Artificial Intelligence. Austin, Texas. p. 53.

Clancey, W.J. (Summer 1989). Viewing knowledge bases as qualitative models. IEEE Expert. pp. 9-23.

Clowes, M.B. (1971) On seeing things. Artificial Intelligence. vol 2: pp. 79-116.

Davis, R. (1980). Meta-Rules: Reasoning and control. Artificial Intelligence. vol 15: pp. 179-222.

Davis, R. (1983). Reasoning from first principles in electronic trouble shooting. International Journal Man-Machine Studies. vol 19: pp. 403-423.

Davis, R. (1983). TEIRESIAS: Experiments in communication with a knowledge-based expert system. Design for Human-Computer Communication. Simes, M.E. and Coombs, M.J. (eds). London: Academic Press.

De Finette, B. (1976). Theory of Probability. pp. 34-36. Wiley & Sons: New York.

Diederich, J., Ruhmann, I. and May, M. (1987). KRITON: a knowledge acquisition tool for expert systems. Special issue on the 1st Knowledge Acquisition for Knowledge-Based Systems Workshop.

Dietterich, T.G. and Michalski, R.S. (1981). Inductive learning of structural descriptions: evaluation criteria and comparative review of selected methods. Artificial Intelligence. vol 16: pp.257-294.

Doyle, J. (1979). A truth maintenance system. Artificial Intelligence. vol 12: No.3.

Dreyfus, H.L. (1968). A critique of artificial reason, thought. Fordham Quarterly. Vol XLIII: pp. 507-22.

Duda, R.O., Gashnig J. and Hart, P.E. (1979). Model design in the PROSPECTOR consultant program for mineral exploration. Expert Systems in the Microelectronic Age. Michie, D. (ed). Edinburgh: Edinburgh University Press.

Earnst, G.W. & Newell, A. (1969). GPS: A Case Study in Generality and Problem Solving. New York: Academic Press.

Erman, L.D., Hayes-Roth, F., Lesser, V.R. and Reddy, D.R. (1980). The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. ACM Computing Surveys. vol 12: No. 2: pp.213-253.

Eshelman, L., and McDermott, J. (1988). MOLE: A knowledge acquisition tool that uses its head. Knowledge Acquisition. vol 3.

Eskridge, T. (1989). Principles of continuous analogical reasoning. Journal of Experimental and Theoretical Artificial Intelligence. vol 1: No 3: pp.179-194.

Fieschi, M. (1982). Sphinx: An interactive system for medical diagnosis aids. Approximate Reasoning in Decision Analysis. M.M.Gupta and E.Sanchez (eds). pp.269-275. Elsevier North Holland: New York.

Fikes, R.E. (1970). REF-ARF: A system for solving problems stated as procedures. Artificial Intelligence. Vol 1: No.1-2.

Fikes, R.E. and Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence. vol 2.

Fikes, R.E. Hart, P.E. and Nilsson, N.J. (1972). Learning and executing generalised robot plans. Artificial Intelligence. vol 3: No. 4: pp.251-288.

Forgy, C.L. (1982). RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence. vol 19: pp.17-37.

- Forsyth, R. (1984). Expert Systems: Principles and Case Studies. p. 14. Chapman Hall: London.
- Gammack, J.G. (1989). Modelling expert knowledge using cognitively compatible structures. MRC Applied Psychology Unit, U.K.
- Glowinski, A., O'Neil, M., and Fox, J. (1989). Design of a generic information system and its application to primary care. AIME. vol 89: pp. 221-233.
- Gordon, J. and Shortliffe, E.H. (1985). A method for managing evidential reasoning in a hierarchical hypothesis space. Artificial Intelligence. Vol 26: pp. 323-357.
- Guzman, A. (1967). Computer Recognition of Three Dimensional Objects in a Visual Scene. AI-TR-228, Cambridge, Mass.: MIT AI Lab.
- Hall, R.P. (1989). Computational approaches to analogical reasoning: A comparative analysis. Artificial Intelligence. vol 39.
- Halliwell, M.A.G. and Lyons, M.H. (1984). The interpretation of X-ray rocking curves from III-V semiconductor device structures. Journal of Crystal Growth. vol 68: pp. 523.
- Harmelem, F. and Bundy, A. (1989). Explanation-based generalisation = partial evaluation. Artificial Intelligence vol 39: pp. 401-412.
- Hayes-Roth, F.J and Lesser, V.R. (1977). Focus of attention in the HEARSAY-II system. Proc. IJCAI. vol 5.
- Heckerman, D.E. (1988). An empirical comparison of three scoring schemes. Proceeding 4th Workshop Uncertainty in AI. pp. 158-169.
- Heckerman, E.J. and Horvitz, E.J. and Nathwani, B.N. (1989). Update on the Pathfinder Project. Proceeding 13th symposium on computer applications in medical care. IEEE Computer Society Press: Los Alamitos. Calif. pp. 721-762.
- Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D. and Slocum, J. (1978). Developing a natural language interface to complex data. ACM Transactions on Database Systems. vol 3: pp. 105-147.
- Henson, R.W. (1987). Model of Image Understanding, Msc Dissertation, University of Warwick, U.K. pp. 5-9.
- Hewitt, C. (1971). PLANNER: A language for Proving Theorems in Robots. Proc. IJCAI-2.
- Hill, M.J. (1986). Unpublished PhD Thesis. Durham University.
- Homa, D. & Vosburgh, R. (1976). Category breadth and abstraction of prototypical information. Journal of

- Experimental Psychology: Human Learning and Memory. vol 2: pp.322-330.
- Homa,D., Sterling,S. & Trepel,L. (1981). Limitations of exemplar-based generalisation and the abstraction of category information. Journal of Experimental Psychology: Human Learning and Memory. vol 7: pp.418-439.
- Kahn,G. (1984). When diagnostic systems want to do without causal knowledge. ECAI-84:Advances in Artificial Intelligence. Elsevier. pp.22-30.
- Kahn,G. (1984). When diagnostic systems want to do without causal knowledge. ECAI-84:Advances in Artificial Intelligence. Elsevier. p26.
- Keene,S. (1989). Object-Oriented Programming in Common LISP. Addison Wesley.
- Keravnou,E.T. and Washbrook,J. (1989). What is a deep expert system? An analysis of the architectural requirements of second-generation expert systems. The Knowledge Engineering Review. vol 4: No.3: pp.205-233.
- Kerry,R. (1990). Integration of expert systems and databases. Report 29, Information Systems Engineering Division. CCTA. Norwich.
- Klatzky,R.L. (1975). Human Memory: Structures and Processes. pp.18-24. San Francisco: W.H.Freeman.
- Klasky,R.L. (1975). Human Memory: Structures and Processes. p.133. Freeman: San Fransico.
- Klatzky,R. (1975). Human Memory: Structures and Processes. p.141. San Francisco: W.H.Freeman.
- Kulikowski,C.A. and Weiss,S.M. (1984). Representation of expert knowledge for consultation: the CASNET and EXPERT projects. In Artificial Intelligence in Medicine Szolovitz,P. (ed). Boulder. Colorado:Westview Press.
- Lakoff,G. and Johnson,M. (1980). The metaphorical structure of the human conceptual system. Cognitive Science. vol 4: pp.195-208.
- Lefkowitz,L.K. and Lesser,V.R. (1988). Knowledge acquisition as knowledge assimilation. International Journal of Man-Machine Studies. vol 29: pp.215-226.
- Legge,D. and Barber,P.J. (1976). Information and Skill. pp.135-136. Methaun and Co Ltd.
- Lenant,D.B. (1982). AM: An artificial intelligence approach to discovery in mathematics as heuristic search. Knowledge-based Systems in Artificial Intelligence. Davis,R. and Lenant,D.B. (eds). McGraw-Hill: New York.

- Lenant, D.B. (1982). Heuristics: The nature of heuristics. Artificial Intelligence. vol 19: No.2.
- Lenant, D.B. (1982). Heuristics: The nature of heuristics. Artificial Intelligence. vol 19: No.2: p.20.
- Lenant, D.B. (March 1983). Theory formation by heuristic search. The nature of heuristics II: Background and Examples. Artificial Intelligence.
- Lindsay, R.K. and Buchana, E.A. and Feigenbaum, (1980). Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project. McGraw-Hill: New York.
- Lindley, D.V. (1987). The probability approach to the treatment of uncertainty in artificial intelligence and expert systems. Statistical Science. vol 2. No.1: pp3-44.
- Lowe, G. (1987) Three dimensional object recognition from single 2D image. Artificial Intelligence. vol 31: pp374-378.
- Loxley, N. (1987). Unpublished PhD. Thesis. Durham University.
- Macrander, E.R., Minami, E.R. and Berreman, J. (1986). Applied Physics. vol 60: p.1364.
- Madni, A.M. (1988). The role of human factors in expert system design and acceptance. Human Factors. vol 30: No.4: p.406.
- Madni, A.M. (1988). The role of human factors in expert systems design and acceptance. Human Factors. vol 30: pp.395-414.
- Martelli, A. and Montanari, U. (1978). Optimization decision trees through heuristically guided search. Communications of the ACM. vol 21: No.12:
- Matsuyama, T. (1984). Knowledge organisation and control structure in image understanding. Proceedings 7th International J. Conference on Pattern Recognition. Montreal. pp.1118-1127.
- McCarthy, J. and Hayes, P.J. (1969). Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence Meltzer, B. and Michie, D. (eds). Edinburgh University Press: Edinburgh.
- McCarthy, J. (1980). Circumscription - A form of non-monotonic reasoning. Artificial Intelligence. vol 13.
- McDermott, D. and Doyle, J. (1980). Non-monotonic logic I. Artificial Intelligence. vol 13: pp.41-72.
- McDermott, D. (1982). Non-monotonic logic II: non-monotonic modal theories. J.ACM. vol 29: pp.34-57.

Meyer, D.E. (1970). On the representation and retrieval of stored semantic information. Cognitive Psychology. vol 1: pp. 242-300.

Mickalski, R. (1983). Theory and methodology of inductive learning. Machine Learning Mickalski, R et al eds. Palo Alto. pp. 83-134.

Michie, D. (1971). On not seeing things. In On Machine Intelligence Winston et al. (eds), New York: Academic Press.

Miller, A.M., Pople, H.E. and Myers, J.D. (1982). INTERNIST-1. An experimental computer-based diagnostic consultant for general internal medicine. Journal of Medicine. vol 307: pp. 468-476.

Minsky, M.L. (1974). A framework for representing knowledge. The Psychology of Computer Vision Winston, P. (ed). pp. 34-57. McGraw-Hill: New York.

Moore, R.C. (1985). Semantical considerations on non-monotonic logic. Artificial Intelligence. vol 25: pp. 75-94.

Murrell, H. (1976) Men and Machines. p. 22. Methuen & Co Ltd.

Mylopoulos, J. and Brodie, M. (1991). Knowledge Bases and Databases: Current Trends and Future Directions. pp. 153-180.

Neisser, U. (1976). Cognition and Reality. p. 112. San Francisco: W.H. Freeman.

Neuro Data. (1991). 444 High Street, Palo Alto, CA 94301, U.S.A.

Newell, A and Simon, H.A. (1963). GPS - A program that simulates human thought. Computers and Thought. Feigenbaum, E.A. and Feldman, J. (eds). pp. 279-296.

Newell, A. Shaw, J.C. and Simon, H. (1967). Empirical explorations with the logic machine. Proceedings of the Western Joint Computer Conference. pp. 218-230.

Newell, A. (1973). Production systems: models of control structures. Visual Information Processing. Chase, W.G. (ed). Academic Press: New York.

Nilsson, N.J. (1980). Principles of Artificial Intelligence. Tiago, Palo Alto, California. Nilsson, J.N. (1986). Probabilistic logic. Artificial Intelligence. vol 28.

Nisbett, R.E. and Wilson, T.D. (1977). Telling more than we can know: verbal reports on mental processes. Psychological Review vol 84: pp. 231-259.

Piaget, J. and Inhelder, B. (1977). The Psychology of the Child. Routledge and Kegan Paul: London and Henley.

Pohl, I. (1971). Bi-directional search. Machine Intelligence 6. Meltzer, B. and Michie, D. (eds). American Elsevier: New York.

Posner, M. I. (1969). Abstraction and the process of recognition. In G. H. Bower & J. T. Spence. (eds). The Psychology of Learning and Motivation. pp. 44-96. New York: Academic Press.

Posner, M. I. & Keele, S. W. (1970). Retention of abstract ideas. Journal of Experimental Psychology. vol 83: pp. 304-308.

Price, C. J. and Lee, M. (1988). Deep Knowledge Tutorial and Bibliography. Alvey Report IKBS3/26/048.

Quillian, R. (1968). Semantic memory. Semantic Information Processing. Minsky, M. (ed). MIT Press: Cambridge, Mass.

Rada, R. (1983). Characterising search space. Expert Systems. Forsyth, R. (ed).

Reggia, J. A., Nau, D. S. and Wang, P. Y. (1984). Diagnostic expert systems based on a set covering model. In Developments in Expert Systems. Coombes, M. J., (ed). London: Academic Press.

Reiter, R. (1980). A logic for default reasoning. Artificial Intelligence. vol 13: pp. 81-131.

Rich, E. (1983). Users are individuals: Individualising user models. International Journal of Man-Machine Studies. vol 18: pp. 199-214.

Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. Artificial Intelligence. vol 5. pp. 115-135.

Sacerdoti, E. D. (1975). The Non-linear nature of plans. IJCAI-4.

Schreiber, G., Bredeweg, D., Dauoodim, B. and Wielinga, B. (Nov 1987). KADS: B2-Design, Esprit Project P1098, VF memo 97.

Shafer, G. (1976). A Mathematical Theory of Evidence. Princeton, NJ: Princeton University Press.

Shafer, G. (1987). Probability judgement In artificial intelligence and expert Systems. Statistical Science. vol 2: pp. 3-44.

Shortcliff, E. H. (1976). Computer Based Medical Diagnosis: MYCIN. New York: American Elsevier.

Shortliffe, E. H. (1976). Computer-based Medical Consultations: MYCIN. New York: Elsevier.

Shortliffe, E.H., Buchanan, B.G. and Feigenbaum, E.A. (Sept 1979). Knowledge engineering for medical decision making. A review of computer based clinical decision aids. Proceedings of the IEEE. vol 67: No 9: p.1216.

Simmons, R.F., (Spring 1986). Man-machine interfaces: can they guess what you what?. IEEE Expert. pp.86-94.

Smith, E.E., Shoben, E.J. and Rips, L.J. (1974). Structure and process in semantic memory: a featural model for semantic decisions. Psychological Review. vol 81: pp.214-241.

Smith, S.F. (1983). Flexible learning of problem solving heuristics through adaptive search. 8th International Joint Proceedings on Artificial Intelligence.

Speriosu, J. (1981). Journal of Applied Physics. vol 52: p.6094.

Stallman, R.S. and Sussman, G.J. (1977). Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit design. Artificial Intelligence. vol 9.

Stalnaker, R. (1988). A note on non-monotonic modal logic' Standard Logic to Logic Programming. Thayse, A. (ed). Wiley: Chichester.

Stefik, M.J. (1980). Planning with Constraints, PhD Dissertation, Heuristic Programming Project, Computer Science Dept., Stanford University, CA.

Takagi, S. (1962). Acta Cryst. vol 15: p.1311.

Tanner, B.K., Chu Xi, and Bowen, D.K. (1986). Characterization of semiconductor materials. Proceedings M.R.S. Symposium. Palo Alto.

Tanner, B.K. (1990). Interview with Authors, Department of Physics, University of Durham, Durham, DH1 3LE, England.

Taupin, D. (1964). Bull. Soc. Fr. Min. Crist. vol 87: p.429.

Thayse, A. (1988). From Standard Logic to Logic Programming. p.149. John Wiley & Sons.

Tjahjadi, T. and Bowen, D.K. (1989). An expert system for X-ray rocking curve analysis. Proceedings International Conference on Expert Systems in Engineering Applications. Wuhan, China. pp.269-275.

Tjahjadi, T. (1990). 3M: A user modelling interface of an expert system for X-ray topographic image interpretation. Interacting with Computers. vol 4.

Ullman, J.D. (1982). Principles of database systems. MD: pp.32-35 Computer Science Press.

- Vere, S.A. (1975). Induction of concepts in predicate calculus. Proceedings of Fourth International Joint Conference on Artificial Intelligence. pp. 281-287. Tbilisi, USSR.
- Wellbank, M. (1983). A review of knowledge acquisition techniques for expert systems. British Telecommunications: Martlesham Consultancy Services, UK.
- Winograd, T. (1980). Extended inference modes in reasoning by computer systems. Artificial Intelligence. vol 13: No.1&2: pp. 5-26.
- Winston, P.H. (1975). Learning structural descriptions from examples. The Psychology of Computer Vision. Winston, P.H. (ed). pp. 157-209. New York: McGraw-Hill.
- Wise, B.P. and Henrion, M. (1986). A framework for comparing uncertain inference systems to probability. Uncertainty in AI. L.N. Kanal and J.F. Lemmer (eds.). pp. 69-83. Elsevier Science Publishers: New York.
- Woods, W.A. (1975). What's in a link: Foundations for semantic networks. Representation and Understanding. Bobrow, D.G and Collins, A. (eds). pp. 35-82. New York: McGraw-Hill.
- Wright, G and Ayton, P. (1987). Eliciting and modelling expert knowledge. Decision Support Systems. vol 3: pp. 13-26. Elsevier Science: North Holland.
- Yen, J. (1989). Gertis: A Dempster-Shafer approach to diagnosing hierarchical hypotheses Comm.ACM. vol 32: pp. 573-585.
- Yen, J. and Jhang, H. (April 1991). Using polymorphism to improve expert system maintainability. IEEE expert. pp. 48-55.
- Zachariasen, W.H. (1945). Theory of X-Ray Diffraction in Crystals. New York: Dover Publications.
- Zadeh, L.H. (1965). Fuzzy Sets. Information and Control. Vol 8: PP. 338-353.
- Zadeh, L.A. (1978). Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems. vol 1: pp. 2-28.
- Zimmerman, H.J. (1987). Fuzzy Sets, Decision Making, and Expert Systems. Kluwer Academic Press: Boston, Mass.

Appendix 1

The Interviews with Domain Experts

This appendix consists of five interviews (see Chapter 6: Section 6.2). Two domain experts from Bede Scientific Instruments, Durham, were consulted. Additional interview material details can be obtained for the Dr. T. Tjahjadi, Department of Engineering, University of Warwick, CV4 7AL.

Interview One Brian Tanner

RH What factors determine crystal quality and how do they influence the appearance of a rocking curve? Perhaps you could name the main ones in order of priority.

BT Ok, if you have tilts and dilations in the film then you get a broadened rocking curve. In things like GaAs on Si the epitaxy isn't good because there is a big mismatch so you tend to get a lot of twining, and that also gives a large range of tilts and dilations so the rocking curve is then extremely broad compared to the intrinsic one, and this is because the lattice is tipped it is sort of tipped sideways like this so that the bragg planes are no longer parallel so there sort of like this, and it won't be uniform like this but a random distribution of tilts, and you also have dilations in there because the effect of mis-orientation on the bragg angle is $\text{minus } \Delta d \text{ over } d \tan \theta$ this is plus or minus the effective tilt.

RH Does that apply to all crystal compositions?

BT That is perfectly general in terms of this broadening which is essentially associated with a micro-mosaic structure. Now if you are looking at a substrate with no film on top, you would expect an intrinsically narrow rocking curve if the crystal is effectively perfect, but if the substrate has a reasonably high dislocation density, it has mosaic regions which are mis-orientated, then that will broaden. The total integrated intensity under the curve remains the same, and that is quite an important point. Hang about it doesn't always.

RH There are exceptions?

BT For very thin layers it does, but for substrates no it doesn't, so it can broaden and also the actual absolute intensity can go up.

RH And that it quite an important factor is it?

BT Well that double axial diffraction is used in the context of CdTe growth, growth of CdZnTe the II-VI compounds, it is used to characterise the CdTe substrates, that width is an intrinsic measure, and similarly in growth of GaAs on Si with very big differences in lattice parameters between the layer material and substrate, then people do use that width as a measure of the lattice perfection.

RH Would the expert be aware of the possibility of the conditions that give rise to that type of effect, would you be able to predict them?

BT Yeh, if I was working on GaAs on Si or a thick layer of GaInAs on GaAs these are layers where the mismatch are several thousand parts per million, then would expect the rocking curve to be broadened because of this effect of the perfect epitaxy. The layer itself would have tilts and defects in it which would give rise to that. Yes so I think if you were an expert and doing work on GaAs on Si for example, you would expect to see a reasonably sharp peak associated with the Si substrate, and you would expect to see a long distance away an enormous broad one from the GaAs, and if it was only a 100 secs arc wide you would be whooping for joy, and throwing your hat in the air for joy, and rushing off to publish the results.

RH Are there any other additional factors that affect crystal quality that would affect the rocking curve? You have spoken about the substrate.

BT The substrate can be broadened by the fact that when you grow (figure 3) an expitaxial layer on the substrate that is a lattice parameter which is not the same as that of the substrate, the substrate will bow. Now that bending means that the angle seen by one side of the beam is different than at the other side of the beam, so consequently the peak is again broadened. Now that curvature can be simulated in RADS and Neil will show you how you can put that in.

RH Is it a similar effect to that of the first effect as in Figure 1.

BT It does broaden it and this time it really does, unless the broadening is really enormous it will keep you with the same integrated intensity, but just pull everything down. One thing you should do when doing this type of analysis is decide which is the substrate peak and which is the layer peak. Now the thing you can't assume is that the substrate peak is always bigger than the layer peak because you may have a very thick layer and the rule is the substrate peak will be narrower than the layer peak, so its width that you go for, and both these effects will not change that criteria.

RH Are there any other affects that affect the quality?

BT That effect the rocking curve?

RH Yes

BT Shape

RH Well a crack for example, I don't know what the technical term for that would be, but damage.

BT A crack in the layer would give you a very broad rocking curve, and almost certainly in an epitaxial layer if you got cracking it would almost certainly look like this mosaic thing (Figure 1) as in micro cracks, because a crack effectively is just a region where the thing has been pulled apart, you got long range strain coming from it. (pause) There are things relating to the way the experiment is set-up which might not be relevant to talk about here which can effect quite significantly the result. (pause) Grading of the epitaxy, in other words if the lattice parameter is varying with depth, will give rise to a rocking curve peak which is asymmetric, and it sort of wedge shaped. That is a characteristic feature.

RH I see, how does that relate to the theoretical concept, I think its called lammellae. Neil tried five for example, he said it might work, but he was not sure how you decide based on the changing composition from the bottom to the top of that layer. How do you decide?

BT If you have a layer that's graded, it depends how big the mismatch is across that grade, a ball park figure I suppose is that you would want one layer per 20-50 ppm, although it depends on how thick the layer is. Usually splitting it up into 20-40 layers, it doesn't make any real difference. So if you cut it into 20 it does really look any different if you cut it into 40.

RH Is there any uncertainty about the composition or grading through the layer when you try to grow one of these structures?

BT From the crystal growers point of view I don't think there is any real way you can know what the composition will be. There is Ga absorption in the case of GaAlAs grown on GaAs. In the case of chemical vapour deposition you can get depletion of one of the elements in the vapour around the growing surface, so as the surface grows you get a change in composition. It is a bit hard to predict whether that will be linear or non-linear. Usually what you normally do is look at the rocking curve and say woops the peaks are not sharp, and they are asymmetric, that looks like a graded layer. Now the difference between a generally bad layer and a graded layer I think is the fact that you would expect to grading on only nearly matched layers when the lattice parameter is pretty close. In other words, you got two peaks and they are relatively close together, a few 100 secs, if the rocking curve is then asymmetric and broadened then you would say that that would be grading.

If you got a very big mismatch, say half a degree away then grading would not a very efficient way of putting that in because you got tilt all over the place. So try to distinguish that broadening between grading and micro-mosaic tilts relates largely to the lattice parameters and what you expect.

RH You know those (lattice parameters) so even a novice would have that information.

BT Well you know what you've tries to grow. It think its very important that if you try to analyse a rocking curve without knowing anything about what it is you are in serious trouble. I've actually had this case from people in the states who say 'we are having this trouble, can you explain what this rocking curve is, its got bumps and wiggles all over the place'. And my reaction is always the same, no I can't, you must tell me what you thought you grew, because the grower knows damn well if he opened the shutter five or six times, and if he doesn't at least its five or six and not twenty two or twenty three.

RH You can't predict the exact number of layers you grew?

BT Well the grower should know exactly the number of layers that were grown.

RH And they would actually be there?

BT I think in all this analysis you have got to try and plug in not data just in relation to the shape of the rocking curve, but also information from the crystal grower. That might be quite important in setting up your system. I personally believe its extremely important..

RH What I was going to do was set up an initial best hypothesis based on a basic screen input of what you think you've grown and follow it up with an interative procedure.

BT I think that is absolutely crucial, there is no way you go in cold.

RH Oh no, I know that.

BT The growers also think they know what composition they've grown, and think they know what thicknesses they've grown, and it depends on your grower and the technique they are using how accurate they can be. I mean if you look at the people at Malvin at MBE if they say they've grown a layer of 160A thick you can be sure that 160A plus or minus about ten isn't far away, with a .1 micron cap and a .1 micron fits the data perfectly, you know. Some techniques not only people but techniques like molecular E-epitaxy??? you can be really pretty confident.

RH Is there a difference between say when you are trying to grow the MQW structure where you've got the A:B A:B layers on top of one another, is there more unpredictability the more complex the structure is or does that not necessarily apply?

BT Not in terms of the reliability of growth data from the grower. You may have diffuse interfaces, but if you are growing an MQW structure you will almost certainly be using

a computer controlled growing kit and the computer opens and shuts the valves the number of times you want it to.

RH Are there any other factors you think affect the rocking curve due to crystal quality? Are there any additional ones you can think of?

BT Well there is also variation with respect to position across the sample. If there is a change in thickness across the sample, and that is really rather fast, then again you can get asymmetry usually in the tails of the peaks rather than any where else.

RH And can you explain, is it interference between the different layers on the rocking curve?

BT The best way to see that is to get hold of RADS and just see that on RADS because they do tend to be rather easy to see. I can give you an example here of these sort of things. These are in fact logarithmic scales. There's a substrate peak which is sharp and then you have, this is a very strongly mismatched layer Figure 4 its InGaAs I'll sketch the structure in just a minute. This is on a logarithmic scale and logarithmic scales are another little can of worms that you might consider. In a way this is quite a difficult structure to start off with. And Figure 5 I'll sketch the actual structure - its .8 microns of GaAs on GaAs and 170Å of InGaAs where the In is .18 and then 0.1 microns of GaAs on top. Now this interference here comes from the GaAs cap, and the relative positions of the peaks is determined by the layer thickness.

RH Is that a method of calculating thickness?

BT Oh yes very much so, you also see on this the InGaAs peak itself which is really rather low, this is log scale, so yes you can use the interference fringes in a number of ways. The simplest way to use the interference fringes is simply to measure the fringe separation. The fringe separation is determined only by the layer thickness. In the case of this particular layer it is the cap layer which happens to be the top layer. If you have a single layer, Figure 6, which is thin, say typically 0.2 microns, and it really doesn't matter what it is, what you will see is the substrate peak, and you will see the layer peak will interference fringes in the shoulders of that peak. Now the separation of those peaks, the thing I'll call θ_p , the pendalosa fringe spacing, $\Delta \theta_p$ goes as $\frac{\lambda}{2d \cos \theta_b}$ which is the cos of the angle between diffracted beam and the inward surface normal divided by the thickness, divided by $\sin 2\theta_b$ where θ_b is the Bragg angle. Now the important thing about that is that it is independent of the strength of the scattering layer, so this will be the same whether its a 0.2 micron layer of GaAs or a 0.2 layer of InGaAs.

RH So its a general effect over all structures?

BT Yeh, and in fact what we have shown is that if you do a fourier transform of this fringe structure you'll get out the layer thickness straight out, and for certainly this A:B structure where you have a capped layer above a mismatched layer on a substrate, that works remarkably well. Simon Miles PhD thesis goes into that in great detail which you might actually look at. now that method of analysis is something that an experienced rocking curve analyst would do almost by eye. You see I would see those fringes and say ah ha yes well those fringes correspond to a thickness of about 0.1 micron. If fact you can draw you self a little table of layer thickness against fringe spacing, and you would put a ruler over the rocking curve and you would not be very far wrong. In fact if you have two layers of similar thickness, you get a set of fringes from each layer. In this particular case of Figure 4 relating to Figure 5 you have one very thin layer with a thicker layer on top. Now this subsidiary peak in the middle here, and this will be an interference fringe, associated with the InGaAs layer.

RH Which layer is that?

BT That's the little bump in the middle here, this is the 170 one. The thinner the layer the longer the period of occilation, and also the weaker the period of the occilation. When there are two together, then you really are in trouble; Mary Halliwell and I published a paper just over a year ago in which we looked at the interference of a structure, the simplest laser structure you can get which was a 0.1 micron layer of GaInAsP and I'm now scribbling this as Figure 8, with an InP cap on, the substrate was InP, so it was literally an A:B:A structure. Now I can give you a copy of the paper. What we found was that if we measured by hand the periodicity of the fringes you didn't get the right thickness, if you just used that simple formula that I have written down below Figure 6. The fringe spacing wasn't inversely proportional to the thickness, there was an offset which we didn't understand at the time, and in the paper we say this - it isn't quite the Bragg case pendulum period. However, what Simon Miles has done subsequently is to take that data and fourier transform it, and when you fourier transform it you get two periods, because in the particular case this capping layer had got very thin and was comparable to the thickness of the GaInAsP layer. So you had two things that were beating together and the eye was being confused. So there may be a strong argument for building into an automatic analysis program an fourier transform to try and pull out layer thicknesses to start with. Those fringes are not affected in principle by grading, in practice they are, and the reason I say in principle is they are there, but the problem with a graded layer is that although the fringes exist the fringe position is sensitive to the mismatch, and so the grading will actually smear out these fringes and the visibility will go down, so if there is a change of lattice parameter with depth through the layer then the fringe visibility can be significantly reduced.

RH And the composition of the substrate, that makes no difference to the fringes?

BT No it doesn't, at all. The substrate composition might affect the way the fringes affect the substrate peak, but no, the fringe spacing is independent of substrate as well. So if you got a multi-layered system with thin layers in and your data is good enough, and there's the rub, you can do a fourier transform and pull out all the thicknesses of the layers independent of the compositions. See you've removed half of your free parameters. That does make the assumption there's no grading through the layers, your assuming its uniform. However, if you have thin layers where these interference effects occur, usually there isn't any grading, because its usually in thick layers that grading becomes important.

RH Is it a rule you get interference below .2 microns?

BT No you can get interference at 1 microns if you look hard. The problem is that the thicker the layer the closer the spacing of the fringes, and so any grading in the layer, a small amount of grading will shift them slightly in position, and if the period is small that will wipe them out. Whereas if you have a thin layer, small amount of grading you'll reduce the visibility, but you will still see the period.

RH Is there anything else you can think of that will affect rocking curve output; crystal quality?

BT (pause) Come back tomorrow I think is the answer

RH Well from the ones that you've mentioned, is there an order of priority, the ones that you would look for first, or does that depend on your initial input or knowledge of what's been grown ?

BT Yeh I think so, if you have a strongly mismatched layer you would expect the layer peak to be broad. It would be very wide and you would expect it to be very wide. You would therefore not expect to simulate in terms of a perfect crystal rocking curve program. That's a starter. And the other important thing is that even though your layer will be poor, your layer peak will be broadened, and because its broadened it will also be reduced in height, because the integrated intensity stays the same.

RH Do the lattice parameters vary according to the type of structure being analysed?

BT The lattice parameter GaAs is very different to InP, so it is specific to the material you are analysing, and the difference in lattice parameter between what you grow on the substrate is going to depend on the composition of the material you are growing. I mean that is the whole reason why rocking curve analysis has become so fashionable because in providing a measure of the lattice parameter of the layer

with respect to the substrate, that difference, with a couple of built in nasty assumptions, you can then get the chemistry. So without doing any wet chemistry so you know for example if you are growing InGaAs or indeed InP you know how much Ga is there.

RH Are some lattice parameters more important than others, so if you wish to establish certain ones is there an order, a priority, because if you've grown a structure you may not have all that information available. Are there some that must be established first?

BT Well the end-members of the binary's, GaAs, GaP, InAs, InP, these are well known, there are well documented. You find them in RADS, there are all there. You assume Vegards law, which is that if you go from say GaInAs, across the GaInAs composition range from GaAs to InAs the lattice parameter varies linearly between InAs and GaAs. If my memory serves me correctly InAs is actually a bigger lattice parameter than GaAs and as the In concentration goes across (Figure 9) so the lattice parameter varies linearly. It's more complicated for a quaternary structure, I can't tell you off-hand what it is, it's a pig of a calculation to do took me about an hour on the train, both for the lattice parameter and the structure factors. Its just like assuming linear variation between binarys, but you actually have four sets of binarys rather than two sets of binarys.

RH Would Neil know about that as I'm doing RADS with him tomorrow?

BT If you start off with a ternary like InGaAs well yes its all sort of built-in. Its all in the RADS manual. (pause) The thing about this type of analysis and these types of materials is that the application is principally to electronic materials. So we are talking about GaAs, GaP, InAs, CdTe, InAt, AlAs, all these lattice parameters are to a better or worse degree well known.

RH And they are all in RADS?

BT They all there in the database. If you want to know what they are go into edit user update database, is it one or two? Just follow the menu, and that will display it for you as well as the scattering factors which determine the strength of the scattering of the layer.

RH This is more general, what types of crystal structure will be analysed by the expert system, the complete range you would wish?

BT If you can do it for cubic structures I'm happy. There are all sorts of cans of worms when you get away from anything other than growth on cube faces, and the reason is quite simply this, if you look at Figure 10, if you have a structure that is cubic which is the substrate of lattice parameter A_0 , and you have another substance that is also cubic of lattice parameter A_L , and you stick one on top of

the other in order to get those to match you have to squash (Figure 11) the top layer in order to get it to fit coherently. What that does is to expand the layer normal to it. Its like tasking a piece of rubber and stretching it, if you stretch it it compresses.

RH Is there a word for that?

BT Well its all related to the poisson ratio, which you may remember from your undergraduate days.

RH Um ah, I'm a psychology graduate?

BT You're a psychology graduate without any engineering, fine. The net result then is that the crystal structure is now no longer cubic, but tetragonal, its distorted in that direction normal to the surface. Now if you start trying to think about how this fits with anything other than cubes it gets extremely messy.

RH So the simulation assumes a cubic structure does it?

BT RADS does at the moment. In the symmetric geometry if actually diffract using diffracting planes which lie parallel to the surface (Figure 12) so that your bragg reflection angle out is same, then you can get away with anything that is not cubic. The body diagonal planes which we designate 111 those would actually work, but this would only work in the symmetric geometry. So it really gets rather difficult and you will; find that RADS in principle is set-up to look at all these possibilities, but in fact most of these are keyed out, you can't actually get in there.

RH Do you need them, are they necessary?

BT As of this moment for the bulk of the users the answer is no. There is a slight little problem that you not only consider a coherent layer, but also the possibility that the layer might relax, go back towards a cubic structure. Now how you determine that is quite difficult because in the purely symmetric geometry like I've drawn in Figure 12, you can't actually determine whether that's happened, because all you are doing is measuring the lattice parameter normal to the surface, so you're actually measuring that expansion or compression due to squashing.

RH Does layer thickness make a difference to that?

BT No it doesn't, not assuming its thin compared with the substrate. Yes if you've got a thin substrate, but for your present purposes no. That correction formula is built into RADS. Its the poisson ratio factor, and if you look at RADS you will also find 'do you want to allow for any relaxation', which is relaxing from what I have draw from Figure 11 effectively to Figure 10 if you stuck the two together. In order to do that you have to neucleate lattice defects called dislocations, and how that happens is part of

a research program we happen to have going. That's not trivial. The way you tell is by looking at reflection that are asymmetric, which so not have bragg planes that are parallel to the surface, so you come in at a small angle and go out at a big angle or vice versa, and by using those, there's your epitaxial layer as well, the planes will be slightly tilted as well. By using those planes you can actually get out the absolute lattice parameters.

RH Next question, what types of range of crystal do you want to grow, like MQW? What are the full range the system will need to cope with?

BT The number of layers you mean?

RH No, more in terms of the general types, is there a classification you can give me?

BT OK, yes you should have that. There's the substrate only I think, which means you've only got one peak. Maybe that's a trivial base class, you can then go in and see how wide it is, which will give you information on the strains on the layer. You then have the single layer, and you want the single graded layer I think which is a variation of the single, then I think you go to non-graded multiple layers in which you consider a few layers, maybe up to about six. I think you then have to go to the MQW which is an artificial crystal, it's a super-lattice which will be A:B A:B A structure, but you must have built in the possibility of mixing that which I guess is your next class is a super-lattice combined with a few layers top and bottom because the grower always puts a capping layer on. Many of these materials will corrode if they don't have a cap of say GaAs. So you got a single layer, a single graded layer, you got a few layers, you got a MQW, and then a combination of a few layers with the MQW. If you are really getting excited then you can consider grading the interfaces, so the interfaces are no longer quite abrupt, but over a few atomic planes, and I really do mean a few atomic planes, so that you can allow for a bit of grading. If terms of MQW that turns out to be important because the effects are adding. If you've got a few relatively thick layers that usually is not that important. As technology is advancing people like yourselves are trying to advance the X-ray scattering techniques in such a way that you get more and more precise and detailed information. If you can't get interface roughness in in the course of your PhD I don't think anyone would fail you on that.

RH For each of those types you've outlined, do they each have a set of typical rocking curves you could specify or recognise?

BT In some cases yes, the MQW is characterised by these little satellites, because it is effectively an artificial crystal, having a periodicity of its own about zero order peak which corresponds to the average composition of the A:B

sequence, you get these little satellites which are equally spaced, from which you can get the period of the superlattice straight out; the thickness of the A layer plus the B layer, that period comes from just putting a ruler across there. So there are ways you can pull these parameters straight out.

RH Are there other characteristics of the other types you have mentioned?

BT The multiple layers, if they're thick and they are different composition layers then you get virtually one peak for every layer, so you see many peaks like this, and again Neil will show you this as he intended to put that on the front of the RADS cover. For thin layers where you've got many layers these interference structure get quite complicated, if you have a single thin layer then you will see a single peak plus the substrate of course, but that has interference fringes on the shoulders. If you have multiple layers, a few layers of similar composition then these interference fringes start interfering with themselves. It's a complicated interference pattern.

RH Do you mean some cancel each other out?

BT Yes, and you find peaks start to split and all sorts of funny things, but this is usually a characteristic of interference fringes and you will recognise a couple of periods there. Again you've got to take care of the data, if the data's grotty then you may not be able to pull that out. Even the expert would not be able to tell and you would need to re-run the data.

RH Are there any others?

BT (pause) I've already alluded to the graded layer having a characteristic wedge structure, it's a broad curve and asymmetric. Come back on that one.

RH Could an expert use information about one crystal type, ie the rocking curve produced from one of those types you've mentioned to help solve another type?

BT Oh yes

RH So if you are working in the dark you could reference a previous example rocking curve to help you solve another problem?

BT Yes

RH If it's a novel structure for example?

BT The different composition will change the position of the peaks because the lattice parameter changes. The different composition will also change the intensity of the peaks because of the different elements there and the X-ray scattering is different. In terms of these lovely little

interference fringes they have the delightful quality that they are independent of the scattering, so yes. You would expect to see those appearing as well.

RH Is there any other information you would use? (pause) Say drawing an analogy between previous one with a different composition?

BT You expect the peak widths to be different, which they would be because of the different scattering strength, so the peak heights and width would be different, but the overall shape I would be analogous. (pause) I think you have to be a bit careful because if you tried compared AlAs with CdTe you would actually be in serious trouble, because CdTe is strongly absorbing and so all these interference fringes would be washed out, and they would disappear because of absorption, but if you are looking at InGaAs on InP as compared to GaAlAs on GaAs then I think it the analogy would be pretty good.

RH I think you've already answered this, but could you summarise this particular question. Are there ranges of features that you can identify for each of the types, ie the MQW, like the height of the peaks, the width of the peak, so I can have a list of features that could be associated with each particular curve?

BT (pause) Lets go to a single layer first, if you got a single layer, the first thing you look at is do you have two peaks. Ok, you've got two peaks, you try and decide which is the substrate that is the one that is narrowest. You would then examine the height and the half width and the position of the peak. The position will give you the composition assuming it isn't relaxed. If it isn't relaxed then it should fit the theoretical model quite nicely, and the half width and the peak height rather well. If it is relaxed then it will certainly be broadened; if it is only partially relaxed. So height, position, width, and then you would look for subsidiary interference fringes to see if they are in the tail of the peak. If you have a few layers which are say an A:B:A structure so you got a layer of different composition sandwiched by two other layers of the same composition, you would expect quite a complex interference structure, and then you would be looking for the interference structure first I think. With the MQW structure you would expect these satellites, you would expect to see a substrate peak plus a strong one, and that strong peak position corresponds to a composition which is the average of the MQW. The average of the lattice parameter A and B.

RH You said that was the zero order peak?

BT That is absolutely right. This is often very intense (zero order peak) not always, I got told off at a short course in San Diego for saying it was always the most intense, but it usually is, and it looks like a peak from a single layer, but then if you look down an order of magnitude in intensity you should find these interference

fringes. If you got a MQW structure it is a good idea to look at it on a logarithmic scale as a matter of practice. These little satellite peaks fall off in intensity monotonically with order, so as you go away from the peak the first order satellites are small and the second satellites are even smaller, and with the third order satellites you are looking hard to see. Now sometimes these satellites are missing and that can give you information about the relative thickness of the two layers (A:B), if for example you have a 1:2 ratio then the third order satellite will be missing. If you look at the satellite peak widths as function of order if you have a dispersion in thickness, in other words there is a variation in the thickness of those layers those will broaden as you go further out. If there is dispersion (Figure 13) so that the layer instead of being say equal thickness AAAAA of composition MNMNMN (I've got those backwards of course) you actually having $A + \delta A_1$, $A - \delta A_2$, $A + \delta A_3$, so that there is actually a variation in layer thickness, as you go further to the largest satellite (in terms of order) do the widths broaden, and if there is compositional grading at the interfaces then you can get satellites asymmetric. So you don't have the positive order satellite the same height as the negative order satellite.

RH Any features you can think of for the other structures, you described the substrate I believe, and substrate with one layer. What about the super-lattices, that is the MQW plus additional layer structures on top?

BT Well there you would see the satellite associated with the MQW, but you would see individual peaks associated with the extra layers, and if they are really thin then you get interference fringes associated with them and it can get extremely complicated.

RH So they could affect the satellite peaks of the MQW?

BT If they were the same composition yes. In these circumstances you would really need to know what your growers thought they grew otherwise you got too many free parameters.

RH Would they avoid growing that type of structure?

BT They might well want to do it. In fact high mobility electron transistors are often grown with a MQW buffer, so on the substrate you grow a buffer layer and then a MQW of say 10 periods and then on top of that you grow another few layers in which the electrons are doing all the work. So that structure is not unlikely.

Interview Two Neil Loxley

RH So this is RADS. Could you simulate a typical single layered structure?

NL Well the first thing you do is tell the computer who you are.

RH That's in the fileheader?

NL Yes the fileheader. User name so type in the user name.

RH It's just a single layer.

NL O.K. so we'll call it single, put that in the database, and enter a comment so we can remember what it is when we come back to it.

RH Incidentally, if you were running a simulation would any of these be numbered, I assume there would be some form of numbering system. Is it a rev number at the end?

NL O.K. the specimen I.D. we put in is TEST which is used to record the data files, so the input files that are created when you describe the sample rocking curve you have dot and then I L and A for the three input files and dot S the simulated rocking curve, and each time you do a simulation the number is incremented so the first one we would do would be TEST.S01.

RH So the maximum is 99?

NL Well it's possible, you might do in an interactive process, but you would probably end-up deleting most of those straight away anyway if they weren't what you wanted.

RH I've noticed on the more complicated structures the time can be up to more than half an hour. Would the user actually have machinery that would reduce that significantly or is that a general run-time for a complex structure?

NL Half an hour would be on a slow computer. I think the longest that's in the manual is a lot shorter than that. Right on a 386 machine, which is what we would recommend most people to use, with a co-processor the longest time took 430 seconds. So that is about 7 minutes.

RH So it is not a problem?

NL Not on a fast machine

RH And there is the 486 also available.

NL No, there isn't actually a 486 available yet, but there will be soon.

RH Shall we move on?

NL O.K., so you press tab to select that (the fileheader option) and then go into create.

RH Right, this is where my questions begin. Why is there an option to exclude the reference crystal from the rocking curve analysis, is this execution time or possibly confusion in the final profile?

NL O.K., the double crystal technique is such that the rocking curve is both dependant on the reference crystal and the specimen. If you want to simulate something exactly you have to include the reference crystal, but to do that you have to simulate the diffraction profile of both the reference crystal and the specimen, and then correlate them together. Now that correlation is about the longest part of the simulation. The only thing that the reference crystal effects is the peak shape.

RH Yes, is that the third process of three stages?

NL It will work out the reflectivity of the layer as you enter the layer, and when you simulate it, it will calculate both polarisation states first and then average them, and then finally it will correlate that with the first crystal.

RH Next question, what effect does the inclusion and exclusion of the reference crystal have on the rocking curve profile of a single layer structure?

NL OK, the only thing it will affect is the peak widths and shape.

RH Not the location?

NL It doesn't affect the location or the splitting of peaks.

RH Now that's general for all structures is it?

NL Yeh

RH OK, Do you think there is any difference in the probability of a user of different levels of experience, ie a novice or an expert including a reference crystal? Are there any conditions where you wouldn't include it?

NL Well for example there are occasions where you wouldn't. If all you are interested is checking your peak splitting then obviously you don't need to, but that's a rarity. There is a time when you would not normally need to use it would be if you were using a synchrotron radiation source. Which is effectively is a plane linearly polarised source of X-rays. Which is exactly what the program simulates.

RH And you can't think of any other instances?

NL Well if you really want a good match you really have to include the reference crystal?

RH But would a novice try to use a closer match when simulating than an expert?

NL I guess the novice would have to use the reference crystal at an earlier stage to ensure that there is a match, because the expert would know roughly the effect the reference crystal is going to have on the shape. So he will be able to predict the outcome when he includes the reference crystal.

RH So obviously that increases the interactive cycle. Yes, I was going to say that assuming that the reference crystal is excluded, I don't know if you can answer this, but when in the re-simulation would it be included? Are there any rules you can think of?

NL I would say as soon as the peak positions are correct, that is when you would start to include the effects of the reference crystal.

RH I believe that in the equipment you supply there occurs both states of polarisation during experimentation. If this is so why are three options available (1 = π , 2 = σ , 3 = both)?

NL Again it is for two reasons, generally in the experiment which most people will be simulating the X-ray source contains both polarisations. (RH unless it's the synchron type). Right, in the synchron radiation source you can select your polarisation states, and that's useful for that. But that's not the main reason, the main reason is the time factor again. Is that enough for you?

RH Yeh fine. Are this is more important. What effect does the exclusion of both types of polarisation have on a single layered rocking curve? Could you deal with each in turn.

NL I guess BT would know more about the exact effect on the rocking curve profiles.

RH Could you hazard a guess. What you believe the effect would be.

NL Well I can say that the dominant polarisation in a double crystal experiment will be the sigma polarisation. That leads to the most changes in the appearance in the profile. The pie polarisation has a lesser effect.

RH And yet the calculation of both in terms of computer time is the same?

NL Yes, so really you would use sigma polarisation, for the first few anyway. It doubles in time roughly when you include both.

RH Does the experience of the user effect the choice. In other words, will the novice user more likely have to

include both types or is the effects on the rocking curve minimal, so you could get any with one?

NL Someone who realised the theory, that with some geometries the pie polarisation was negligible, because there is a $\cos^2 \theta$ factor in there.

RH Could you right that down?

NL (Figure 1) The polarisation factor is C which is used in the dynamical theory, and it's a simple multiplying factor. Now C equals $\cos^2 \theta_B$. θ_B is the Bragg angle. So you see if θ_B is around 45 degrees the polarisation factor for pie polarisation goes to zero, but for the sigma polarisation C is always equal to one, and that's because in the sigma geometry the absorption or attenuation of the X-rays is independent of the angle.

RH Would a user using the system always know that do you think?

NL Well I think (pause), not necessarily. Certainly if you are using bragg angle close to 45 degrees then there is no point in including pie polarisation because it will always be zero or close to zero.

RH And I suppose if they didn't know they would always use number two option (σ polarisation) anyway.

NL Yes

RH Are the $\text{CuK}\alpha_1$ $\text{CuK}\alpha_2$ wavelengths typical values used by the system? That's the standard you install. Is that generally what all labs use?

NL $\text{CuK}\alpha_1$ is the most commonly used radiation source for experiment. The double crystal techniques, such that the $\text{CuK}\alpha_2$ is included in the diffraction profile, but the experiment is not wavelength dispersive, so the effect of the width of the spectral divergence of the X-ray beam does not show up in the experiment, because the experiment is non-dispersive in wavelength.

RH What effects does the change in wavelength have on the rocking curve profile? So if you used Mo, and used the same sample, same conditions, but simply changed the X-ray source?

NL The shorter wavelengths will tend to reduce the peak widths.

RH Will that compress the whole scale over which the rocking curve is spread?

NL Yes that will change the full width half maximum over of the peaks, and because the angle at which things diffract is dependent on the wavelength as well as the wavelength goes

down the Bragg angle will go down, and consequently the splitting between the peaks will go down.

RH Can you predict in a formula the amount that it will reduce?

NL Well it's related by the (Figure 2), well it's just Bragg law really. $\sin \theta$ equals λ relates to the wavelength to the angle. There is a formula for half widths of peaks, a fairly simple one, which I don't have on me; it's something BT would have.

RH Would a user lacking in experience be confused by the modification to wavelength or is that just an impossible situation?

NL (pause) Well generally he would know which wavelength he was using. Again, synchrotron radiation source, where you have a tunable wavelength, then if he has a well defined sample it could be used to determine the actual wavelength of the sample he was using.

RH Is there a typical set of Miller indices used from the specimen?

NL Yeh, generally a symmetric reflection is used from the 001 orientation surface. The most common one is the 004 reflection. Now the reflection indices are not Miller indices. Reflection indices are the Miller indices multiplied by the order of diffraction. (Figure 3) So if we put the reflection indices in full type HKL, and we say that the Miller indices are denoted by italics *hkl*. Then *h* is going to equal nH , *k* is going to equal nK , and *l* equals nL , where *n* is the order of diffraction.

RH (pause) How do these indices relate to crystal composition, ie are there any rules that suggest the use of a particular set of indices for a particular composition, and does it vary depending on the sample you have got?

NL (pause) Right, most of the III-V compounds which is generally what's looked at here, then you have a 001 surface. Sometimes you'll come across 111 orientation substrates, for example you might want to simulate the effects of Si with a 111 surface, which will affect the reflection you are using. You may normally use a 333 something along those lines. Generally, for a symmetric geometry you wouldn't normally use a different reflection just because you were using a different material.

RH So there's no relation?

NL You might use a different asymmetric reflection depending on the absorption of the layer, and how far you want to penetrate into the crystal.

RH Does the type of sample available to the experimenter restrict in any way the selection of those indices?

NL Yes, because like I said most III-V's are cut with a 001 substrate orientation. So, therefore, if you want to use different reflections you are going to have to start using asymmetric reflections, which again we may come to later. In terms of symmetric reflections then, yes you will be limited by the orientation of the substrate.

RH I believe that a cubic structure is the only one modelled in RADS. Does that present any problems to the experimenter?

NL Not at the moment, there are plans to include different structures in the future, but I would guess that 90% of users would be using just cubic structures.

RH Is that what the samples would be, or would they just be modelling a cubic structure?

NL No, the samples they will be modelling will all be cubic.

RH But could you model a sample that is not a cubic, and just get away with it?

NL (pause) You wouldn't be able to even start doing that because you need to enter its structure, the structure and composition of the material before the program can start to do a simulation.

RH The scan range is variable. How does composition of the sample ie a single layer dictate the scan range? Could you give me rules on that, what an expert would use.

NL Well again this goes on experience and maybe BT has some more precise figures. The composition moves the layer peak linearly with composition. So it depends on the structure again, the type element you change the composition of, thus by which direction and by how much.

RH Sometimes it's on the right of the substrate peak and sometimes on the left.

NL Left and right, it depends on whether the compound has a lattice parameter which is smaller or greater than the substrate.

RH So which is which?

NL (pause) It's just Braggs law. Its $2d \sin \theta$, but d is actually $a / \sqrt{h^2 + k^2 + l^2}$. $\sin \theta$ equals λ , so if a goes down then θ must go up to compensate for it. So if the lattice parameter of the layer is greater than that of the substrate it will be to the left of the substrate peak, and if the lattice parameter is smaller than the substrate it will be to the right of the peak.

RH And I assume that the user user would know that.

NL Generally it's crystal growers who will be using the program and are much more aware of the affects of lattice parameter of different doping materials or different compositions.

RH When initially simulating a structure would you select the same scan range as used by the experimental set-up if you are simulating, and if not why not? And how does that change when you iterate down to a solution?

NL (pause) Well if you have an experiment you are trying to simulate then you can see from the experiment where all the significant feature of the rocking curve are. So you would try and mimic the simulation to same range as the rocking curve that has anything significant in it. So if you had a large amount of background on either side, for example, you would not include that on the simulation. You want to end up with something that uses the same scan range as the experiment. Generally an experiment would be done so that it only include significant peaks of the rocking curve, so you would not normally change that. (RH Oh). Well you might do if you are just concentrating on getting the composition of one of the peaks. You may just concentrate on simulating around a certain area so include the substrate and one layer peak if there is more than one peak.

RH Is there any order, would you tend to want to find out about the layer or the substrate first?

NL Well the substrate will always be at zero. So it's the layer that's going to affect the splitting. I can see if you are doing a super-lattice where you get satellite peaks that are a long way out from the substrate peak, then you might want to find out what the average composition of the super-lattice first. Now you would do that by concentrating on a area very close to the substrate peak and forget about the satellite peaks. Well maybe not, the position of the satellite peaks will be dependent on the period of the super-lattice which again will affect the average compositions. You would be able to do much before having to include most of it.

RH It's possible to change the scan step. For a simple structure what is the typical level of resolution that you would use for simulation?

NL Well eventually you would try and match it down to what your experiment is to get the same resolution as the experiment. Generally I would say a typical scan step would be about 2 arc seconds, but you can use something that is courser than that to get rough details.

RH If you choose a very large scan step I assume you can loose a layer could you?

NL It is possible to loose a layer, although evidence of it would probably be there.

RH Is that where you might use the logarithmic option?

NL No, that's not true. It's is not so much that you might loose a layer, it's because of the truncation of the layer due to the large steps, you will actually get something that is a strange shape.

RH Right, so you could actually have something with an asymmetrical when in fact it's not.

NL Yes that's right, because you may just get two steps on one side of the peak and only one on the other, which would make it look strange.

RH There are twelve substrates in the database. Does that cover all the options?

NL No it doesn't, but it's possible to enter any substrate that you want to. That's a separate command to add to the crystal database. The program already includes all the information it needs for all elements.

RH Are you talking about the simulation program?

NL I am talking about then simulation program. There is an elemental database which contains all the scattering factors for all the elements that they are known for. All you need to do when you add a new material for the substrate or a layer is to specify the structure of the crystal. That is the type of structure and the lattice parameter, where the atoms are in the unit cell, and once you've done that the program can work out the X-ray scattering for that particular structure.

RH Would a user be more familiar with certain rocking curve profiles produced from a certain substrate? Are there any typical ones that are always used or does that vary from lab to lab?

NL Really that depends on what they are growing. You find a lot of places and people only have experience of growing one type of material, and they would be very familiar with what that looks like.

RH So there are no probabilities that you could assign to that?

NL No, not really, it varies more from lab to lab from what they are growing and what they are trying to do.

RH And that could cover the full range?

NL Yeh, they grow all sorts.

RH If a new substrate is grown that is not included in the database could you use another substrate structure that is similar if you didn't know how to add or you didn't know the composition of it?

NL Yes, there are some matches. For example, Ge is very close to GaAs. You could do it by finding something with a similar lattice parameter, but that is not the way to do it, the way to do it is to add a new material to the database.

RH So in theory you should always be able to add the new material no matter what?

NL Yes.

RH Will the reflection geometry be known to the user given the experimental conditions?

NL Generally speaking yes, once he understands the wording on the screen and how that relates to how he set-up the experiment then he will know the reflection geometry.

RH Is there a typical reflection geometry used for substrates?

NL Well you can't really devolve the substrate geometry from the layer geometry as they are the same. Typical would be symmetric which is the most commonly used on a routine basis. The next one to be used would be the asymmetric glancing incidence, there are not many people using the asymmetric glancing exit at the moment.

RH Where would each type be used?

NL Well symmetric geometry is the one that is easy to use and would be used as the routine measurement. The asymmetric geometries are used to limit the depth penetration of the X-rays, so you can bring up layer peaks relative to substrate peaks for instance.

RH Would that be used for very thin layers or complex structures?

NL Yes, if you have a very thin layer you have to use an asymmetric reflection to be able to see the layer peak. (RH How thin?) Oh, you are talking about sub-micron 0.1 micron and below.

RH Would that also relate to multiple layers like MQW?

NL Because there are multiple layers the reflecting power of the MQW structure is very large anyway.

RH So it's only when it's a very thin individual layer on top of a thick substrate that it matters?

NL Thin layer full stop! It doesn't matter if it's on a thick substrate or between individual or not. What you would be trying to do is reduce the effect of the substrate in relation to the layers on top, coming in at a low angle so the X-rays don't penetrate so deeply into the crystal.

RH What's the reason for the glancing exit then.

NL I can't remember BT might know KB. You actually get the same depth of penetration with the glancing exit as you do on the equivalent glancing incident because with X-rays you can reverse the path. The other reason you use asymmetric reflections is to measure the lattice parameter in a direction parallel to the surface normal.

RH Is there a formula for that?

NL (Figure 5) On a symmetric reflection the only lattice parameter that is influencing the diffraction, relating it to the substrate diffraction is that which is parallel to the surface normal. So for a symmetric reflection you are only picking up lattice parameter (lattice spacing) parallel to the surface normal. Now most layers are grown so that they are coherent so that in effect the layer is strained and you get what is called tetragonal distortion, which is an upward distortion, and that accounts for the mismatch that you measure. Mew is the poisons ratio, which relates to the way a material reacts when you squeeze it in one direction ie the extent the material will move in the opposite direction (the tetragonal distortion. So when you actually a mismatch you are measuring something that is half. The mismatch of the layer before you put it on the substrate which is what you are interested in because that tells you what the composition is, is related to the effective mismatch which is what you can measure. The effective mismatch is M^* and you multiply it by one minus the poisson ratio over one plus the poisson ratio which gives you the real mismatch ie the mismatch before you put it on the layer. Now that means in general that the effective mismatch is about twice the real mismatch. Now if you want to know how much the material has relaxed ie the layer is not so much strained (RH gone back to its original shape) Yes, then you need another measurement that will pick up the lattice parameter parallel to the surface of the crystal, and that is where you would use an asymmetric reflection where the lattice spacing that you pick up is both a mixture of both the lattice spacing parallel to the surface normal and parallel to the surface.

RH What is Vergard's Law?

NL That just relates mismatch to the composition.

RH Actually I haven't covered that, could you cover that now?

NL Say you have an AlGaAs layer you can relate the composition of the Al to the mismatch that you measure by something called Vergard's Law, which is where you extrapolate between the lattice parameters of say the constituent parts, which would be GaAs and AlAs so you have the lattice parameter of GaAs and AlAs and you measure a mismatch which tells you that the lattice parameter of your layer, and because you know the lattice parameter of your

layer you assume a linear relationship between the lattice parameter and the amount of Al in there. So you extrapolate between the values of the lattice parameter of the constituent parts.

RH So that does vary depending on composition?

NL Yes. (RH a formula) I don't have it on me. BT will fill you in on that one.

RH OK I'll move on. When cutting the wafer can the misorientation be determined prior to the experiment, if not how is this determined?

NL Yes you can measure the wafer misorientation very easily. What you would normally do is you would rotate the sample about the surface normal and measure the displacement of the Bragg peak as you do so. At the extremes, if the layer tilt is parallel to the plane of incidence, then between 0 degrees and 180 degrees you will measure the peak shift of twice the layer tilt. So you measure the position of the Bragg peak for 0 degrees and for 180 degrees, and you know the substrate tilt is half way between.

RH Do you envisage any difference in the ability of users (expert/novice) to determine reflection geometry?

NL No, it's easy. Once the geometry is explained to them then it's obvious which geometry has been used.

RH What is the skew angle?

NL Its not in the first release (RADS). Normally when you do an experiment you arrange for the diffraction vector to be in the plane of incidence, and that removes all effects of dispersion. If you choose to have the X-rays incident, so that the diffraction vector is actually lying outside the plane of incidence, you can compensate for that by reducing the Bragg angle or increasing it slightly, but that leads to dispersion from the crystal which will lead to peaks that are broader than they should be, but in doing so you can also do something very similar to what you do on a synchron radiation source which is continuously tune the depth penetration of your X-rays, because you can continuously vary the angle of incidence of your X-ray beam whilst still satisfying the Bragg condition.

RH (pause) There are twelve reference crystals in the database, the same used for substrate. Will the experimenter use only the crystals from this list?

NL Not necessarily, but again he can enter his own reference crystal if it's not there.

RH Why is there a selection of reference crystals I thought they were fixed?

NL Again, to get rid of wavelength dispersion you have to match the lattice parameter or rather the d spacing, that's the inter-planer spacing if you like, for both the reference crystal and the substrate.

RH Is there a figure you could draw?

NL (Figure 6) What you are trying to do is you are trying to make the Bragg angle of the first crystal equal the Bragg angle of the second crystal. That means that all wavelengths are simultaneously diffracted. All wavelengths that are diffracted by the first crystal are diffracted by the second crystal in the same angular position, so as you rotate the crystal you don't select different wavelengths. You would choose your reference crystal to match that of your substrate (not the layer).

RH Are all the parameters of the reflection geometry known at run-time?

NL (pause) Yes normally.

RH And if you didn't know them you would be in trouble?

NL Again, you would get reasonable matches, but the peak shapes would be skewed.

RH Are there any experimental conditions where that might show up as an accident. Could you detect that if you set the experiment up incorrectly?

NL Yes I guess you could. You could try different reference crystals and see what effect it has on the profile, but it's unlikely that someone would not know that there is something wrong.

RH Can you trick the simulation program into producing better results by using a different reference crystal, like as a cross comparison between one and another?

NL Not really, you wouldn't normally use it in that way.

RH Would you ever simulate just the substrate and miss a layer out?

NL Yes, simulating just the substrate is very good for finding out what the theoretical half widths of materials would be because many crystal growers use half widths as a measure of crystalline quality.

RH Can you show me what the half width is with a figure?

NL Well that varies with material, obviously, which is one reason why it's nice to be able to simulate it (Figure 7) You are trying to relate the full width half maximum. As an example GaAs is of the order of seven arc seconds theoretical. The worse the material the broader the rocking curve will be. So once the experimenter has determined its

theoretical minimum is he can judge how good and bad the quality of the substrate is. For example for Si it's possible to obtain perfect Si, so you would expect to be able to match the theoretical value exactly. With things like GaAs and InP it's possible to get close, but at the moment the growth process isn't good enough to eliminate all the damage of the crystal.

RH Right, and are there tables of this information?

NL The people pick-up values for, theoretical half widths from various places and it's never been put together.

RH Unfortunately I'd need that information.

NL Well this is where the simulation program would be used to determine this. There's a simple formula, which is in BT's book which relates the half width of a crystal to various parameters, but that will only work assuming a single crystal.

RH So the simulation program is a way of producing that?

NL Yeh, because you have to include the correlation effect as well.

RH How does the thickness of a single layer affect the rocking curve?

NL The thicker the layer the greater the integrated intensity of the peak. As the layer gets thicker the integrated intensity of the substrate will go down because more of the X-rays will be absorbed before they can be diffracted from the substrate. Above a certain thickness layer cannot be coherent, the strain cannot be accommodated in the layer, so the layers tend to relax when they are too thick, which means you no longer have the effective relationship between mismatch and effective mismatch.

RH What does that result in (a very thick layer incoherent)?

NL That would move the peak. It would have a broadening effect, but that is a secondary. If the layer relaxes then the peak will move, and by simulating it you can measure the degree of relaxation of the layer provided you are confident of the composition you had grown.

RH When you spoke of the integrated intensity, is the peak more likely to broaden or go up in intensity as the layer gets thicker?

NL It goes up in intensity as long as the crystalline quality remains the same, so it doesn't make any difference to the half-width.

RH Is this layer thickening effect general for all compositions?

NL Yes. Some compositions are more highly absorbing than others (pause).

RH Can you give some examples?

NL No, not off the top of my head BT would know, but it can be said that the thickening of some layer compositions has a more dramatic effect on the substrate peak than others.

RH If the structure is graded what determines the number of lamellae chosen?

NL Really it's the rate of change. You have to try and match the thickness of the lamella to be so that the composition is not changing significantly from the top and bottom of the layer. I don't have an exact number for what that should be. Again BT has the experience to do with this. What you are doing though is splitting the layer into lamella of constant composition.

RH Which is a theoretical construct?

NL Yes.

RH Does that effect (lamellae) vary across the compositions you are using?

NL No.

RH I believe a biaxial strain is introduced to the surface when the layer is fixed to the surface of the substrate. How does that affect the rocking curve profile?

NL I think I've answered this one already. That's coherency and tetragonal distortion.

RH Does layer thickness combined with layer relaxation interferes with the rocking curve profile as the layer gets thicker?

NL Yeh, I think we've done that one as well. Layer thickness is one of the things that causes layer relaxation. If the layer gets too thick that it can't accommodate the strain then it has to relax at the interface. (RH so BT would know more?) He has much more a feel for the figures.

RH Is the user expected to know how to calculate layer relaxation?

NL There is no hard and fast way of calculating it. What the program allows you to do is enter a percentage value, so that a 0% of relaxation is a fully coherent layer, which means that the lattice parameter parallel to the interface of the layer is the same as that of the substrate, and fully relaxed is 100% where the lattice parameter is actually equivalent to what the layer would be if it was not deposited on the substrate.

RH How do you know what amount to put in then?

NL Well you would normally expect something to be either relaxed or not relaxed. Then you can actually use the layer relaxation to tune the rocking curve to establish how much relaxation is there.

RH Is that one of the last parameters you adjust?

NL Not necessarily, because it's affecting the peak position as well as the peak half widths to a lesser extent. You would normally try and obtain a good match for the peak positions first, before worrying about things like grading, exact layer thicknesses and so on. So peak position is quite an important one to get first, and relaxation will affect that, but generally people grow structures where the layer is fully relaxed or fully coherent.

RH End members have a constant composition, whereas compounds do not, I assume that endmembers cannot be graded. Is that right?

NL That's correct.

RH Could you show me how you define the X composition parameter (from RADS)?

NL OK, (Figure 8) X is equal to $At^2 + Bt + C$, and t is the position from the bottom of the layer. Now that means that you define A , B , and C to determine the composition of the layer and the composition grading, so if the layer has a constant composition you only need to define C , and X end up as a fractional composition. So, for example, if there is 40% of that particular material you would enter 0.4. Linear grading you would enter a value for B and you would set A as zero, which means that as you go through the layer the composition increases linearly. If you wanted it to decrease linearly you would have to enter a negative value for B . For quadratic grading where you have At^2 composition then you don't have to add a linear or constant component. You can have any one of those constants on there own.

RH Are you able to state what a typical grading would be or is that open-ended?

NL Graded layer are usually used to accommodate a layer which has too high a mismatch between layer and substrate. So graded layers would be used to accommodate the strain between two layers, the layer and the substrate or between two layers, so that the layer doesn't relax. So you would use it to go from one composition at the bottom of the layer to another composition at the top.

RH Does the experimenter know what the grading will be at the start of an experiment?

NL He knows what he hopes it will be. That means he hopes that he's grown it in a certain way. So if his machines

accurately calibrated and everything is working properly, then the composition will be what he thought it was, but that is not necessarily so.

RH Is there a percentage on that, is it a common occurrence?

NL Well this is one of the things, that as the crystal grower get experience they improve at, so they are actually growing what they think they are growing, and the simulation would be one way of checking that. They may grade layers especially to check that they can grade layers, because some of the graded layers would be too thin to pick up individually anyway.

RH Is there a control condition that they would use?

NL I don't know. It's a crystal growers domain really.

RH Are some compound more likely to be graded than others?

NL yeh, you would grade things where you are trying to achieve a high mismatch in your layer, so you need to accommodate that. Some people actually grade device layers, but I'm not sure why. The device layers are those that actually do the work, the other layers are buffer layers, there to achieve a certain lattice parameter, and to achieve a dislocation free perfect active layer. The active layer has to be the best.

RH What effect does a change in composition through the sample have on the rocking curve profile, ie could you identify the rate of change through a layer from the rocking curve?

NL The layer of composition grading is normally asymmetric. If you look at it you will see that one side is wider than the other, and the grading will affect the shape of it more than anything else. That's about as far as I can go.

RH What do the Y composition parameters refer to?

NL Y composition parameter is used in quaternary compounds where there are four elements, and indeed for four end-members, four constituent end-members. So if you have something with four end-members it's possible to change the composition of two elements in that structure.

RH The same question for these quaternaries. Does it also produce an asymmetric peak?

NL Only if it is graded. The effect is the same as quaternary accept you have two variables.

RH Is there a difference in the rocking curve profiles or not, is there more asymmetry when you are grading two elements?

NL Pass, I don't know.

RH Would they do that?

NL I don't know. The reason for the quaternary is so they can tune the band-gap to get a particular wavelength.

RH Is this a big area, band gap engineering, is it something I should cover?

NL Well I would say that a lot of people are using our instruments to study these things. What you can do is tune LED's to emit a certain wavelength by changing the band gap, and one way to do that is to change the composition of the material. Quaternaries are used because you can much more finely tune the band gap.

RH And a lot of user are using your equipment to do that, what percentage?

NL About half (50%).

RH The plot command has both a linear and logarithmic option. When would you use linear and when would you use logarithmic?

NL Linear plot is how your data will be recorded, although it's possible to show experimental data in logarithmic mode as well. Linear mode you would use where there are well defined peaks, with good intensity and no fine structure to speak of, ie no interference fringes. As soon as you get anything that is in the 1% level or less you really need to go to the logarithmic mode to show the details of the lower intensity in preference to those of the high intensity.

RH When using analyse three sets of calculations, calculating the s polarisation, the reference, and correlating with the reference are performed. Can you describe briefly each of these?

NL There will be more steps depending on if you use both polarisations. When it is calculating s polarisation what it is actually calculating is the reflectivity of the crystal for the range of wavelengths which are allowed by the instant scan range. So it is working its way through the substrate in the crystal for each angular position. So it goes through the layer and then substrate lamella by lamella, and works out the reflectivity for that angle and then go to the next angle and work out a profile.

RH Is it possible to have a diagram of that?

NL (Figure 9) So if you imagine you has a rocking curve that ended up looking like Figure 9. That's your scan range you put in -100 say to 100. What it would do is for a substrate and a layer which is split into two lamellae, would say for -100 for theta equals theta B minus 100, it would start at either the top or bottom, not sure which, say top, work out a reflectivity then go downuse what it has calculated for

the first one and use that for the next part as well. It's an iterative process and works its way down the crystal for that value of theta. Then it would go onto the next value of theta which is -100 plus the step size, and it will do it again. So for each point it has to go right through the layer and calculate the reflectivity.

RH Now the reference?

NL The reference is doing exactly the same sort of thing, but for the reference crystal.

RH The correlation?

NL It then correlates them with the reference which means that it slides one over the other, and where the intersection points are it multiplies one by the other so you get the effect of the reflectivity of both crystals combined as one is rotated past the other one.

RH Right that's it then no more questions.

Interview 3 Brian Tanner

RH Is it more or less likely that the user will exclude the reference crystal if the structure is complex?

BT (pause) In using a simulation program yes, obviously not in the experiment, the reference crystal has got to be there, but yes because a very large amount of the computational time, as you will have discovered already is taken when you convolve these two data sets together, and if you are looking at MQW where the angular range of the peaks when you include the satellites is very wide then that is going to take one heck of a long time.

RH Are then any values you can put on that? Are there any types of structures where you would exclude or include it to save time on re-simulation?

BT OK, you can only get something like 3000 points on RADS. Now if you take steps bigger than about five arc seconds then your peak shapes start becoming not representative of the real structure, and you lose information. So if you go over a range of more than 5000 or 6000 seconds then that's is the time you exclude the reference crystal.

RH I assume what you're also saying is that if the structure is very complicated you have got to use very small steps when simulating to get enough detail?

BT Arrr yes, it may even in fact it may not even be a complicated structure it could just be a highly mismatched structure where the peaks are a long way apart.

RH Or if there is interference in the structure?

BT Usually that's not too much of a problem because the scattering falls away quite sharply, it goes roughly as the angle to the minus four which potentially goes up inverse fourth power. So it does fall off very fast. Now the interference modulates that so by the time you've gone a 1000 seconds away from the Bragg peak, even if you have interference effects on there, unless you've got another peak coming up, the actual intensity is really very low.

RH Assuming that the reference crystal has been excluded, when in re-simulation would you include it?

BT When you got the structure right.

RH What do you mean?

BT When you are satisfied that the match looks good apart from peak widths. It will be relative peak heights that you have settled on and interference structure and position of peaks.

RH What type of polarisation would you include for a complex structure, and is that any different from a simple structure?

BT The situation regarding polarisation is no different for simple or complex structures. The polarisation affects the physics, the scattering is different for the two polarisation states. IF you have a regular X-ray generator then there is unpolarised radiation coming from it, and unpolarised radiation can be considered as two plane polarisations at right angles, but with random phase between the two states. So the way in which we handle it is to calculate for the sigma polarisation and then for the pie polarisation. You do it for both and divide by two, so you are averaging the two polarisation states. I'll draw you a diagram (Figure 1) of what I mean by these polarisation states, there's the scattering plane, there's the beam in and there's the beam out. When the polarisation vector the electric vector is normal to the scattering plane we call that the sigma polarisation, and if it's in the scattering plane we call that the pie polarisation. That's exactly the same for standard optics and you may have come across that at school. Now the actual intensity scattered in these two cases is different so the widths of the rocking curves is different, and the heights of the peaks is different. I have had a discussion with KB about this, and I'm quite clear that to do the simulation properly you have to do each simulation independently and add the final result because you are then adding the results incoherently.

RH Are you saying that you simulate first with the pie and then with the sigma and add the two together?

BT Yes, and that what option three is. That is done automatically for you. And that corresponds to work in a regular laboratory with a regular X-ray source. It does matter, in testing RADS KB and I had quite an interesting week in which we did a cross talk act, and it was the fact that he had assumed only one form of polarisation and I was doing both, and the results were really quite different.

RH Can you elaborate of what actual difference these two types (polarisation) will make for an MQW structure?

BT The first effect is very clear, the half width of peak changes, the intensity of the peaks change.

RH In what direction?

BT Pie will always be smaller in width and lower in intensity. Your sigma is your base line. That is something I think many users will not be aware of. The important thing is that the difference is a scattering polarisation term which is the cos two time the scattering angle, that's two theta (Figure 2), this factor the cos is often at about .9 unless two theta is really getting very big. So for many, for example III-V semi-conductor with standard 004 geometry, which is the one most people use in the laboratory with Cu

radiation, then the difference is very small, but if you use 90 degree scattering angle then it can be really quite significant.

RH Will the exclusion of a certain type of polarisation confuse a user during re-simulation?

BT For standard geometry 004, GaAs, 1.45A it won't matter. If you go to the 044 reflection, glancing incidence then the scattering angle is close to 90 degrees and the cos of 90 is pretty small. If you only use the sigma polarisation, well I'm not sure if it will confuse, but you just won't get a decent fit.

RH Is there any reason to change the wavelength of the X-ray, a synchron source, when analysing an MQW structure?

NL suggested this was used for cross comparison for different depths in the crystal.

BT Yes, as you change the wavelength so you change the penetration of the X-ray waves into the crystal, and so that can give you more or less surface sensitivity. That will happen because of both the intrinsic scattering, but also because of absorption, these two effects. Yes people will use synchron radiation as a variable parameter.

RH And is that a common technique?

BT Very few people in the laboratory change X-ray tubes. Cu is a standard X-ray tube because the targets are easy to make, you don't have to do any plating or anything, but they tend to be the most robust, the ones you get the most power out of, and the net result is that most people have that set-up. There are a few people I can think of who use Cr radiation, but very few and far between.

RH What about Mo?

BT Mo is used for work on Si where you want very high sensitivity, but for epi-layer work such as most people use double axis diffractometers for.

RH What about Ag that is slightly longer?

BT Shorter. Ag is 0.5A, Mo is 0.7A (RH Oh NL said it was longer) Don't believe what all my students tell you. I'm not to be quoted on that tape recorder.

RH It will all go on the transcript.

RH Is there a particular scan range over which you analyse an MQW structure, given a wavelength of 1.45A?

BT No, it depends on the period of the super-lattice. If you like to think of it as the ultimate limit, if you only have two atomic spacing for each layer then the splitting between the relative satellites will be comparable to all the bragg peaks in a real crystal, because the structure is only twice

the height of the intrinsic structure of the crystal. If you have a very long period, MQW structure, say 100A and 100A, the peaks will be very close to the zero order peaks. So the answer is no there is no real figure you can get. You would not normally expect to go over a few degree, and if you did you would go to a powder diffractometer anyway.

RH Do you use a particular scan step for an MQW structure if simulating?

BT The problem with MQW's is that the peaks are narrow, but the peaks are weak, so in collecting the data it's a long experiment so the scan steps have to be small. It just happens to be a topical point on which our users are coming back to use about.

RH Is the a typical substrate upon which an MQW structure is grown or does that depend on the lattice match ie the potential for lattice relaxation?

BT I can tell you on what most MQW's are grown, but the answer is that crystal growers will try and grow anything on anything. So you can't assume that some idiot won't try and grow something incompatible.

RH Well the reason for the question is to try to place constraints on the process so you limit the amount of search.

BT Fine. For the present time ie from 1990-91 you would reckon that GaAs, InP, Si, and CdTe would be the substrates. Now the MQW structures well on CdTe it would be CdHgTe with CdTe, or CdZnTe, so it's a mixture of those on which you got ternaries. On Si it would be principally SiGe so the period would be Si the SiGe. On the InP it tends to be a quaternary of InGaAsP and usually its mixed with InP. On GaAs you are principally looking at GaAlAs with GaAs, and sometimes a pair of binarys AlAs, GaAs, AlAs, GaAs, and so on. You should look for those first.

However, as control gets better at the molecular level you might start seeing thing like GaAs InP periods. I would have thought the grower new what he tries to grow so I would not see that as a free parameter.

RH Well it's just you use that for constraining the reasoning of the system.

RH Would the experimenter a symmetric reflection orientation?

BT In the first instance yes, but if you got significant relaxation no. So if you are concerned about measuring relaxation, say SiGe on Si, you would certainly use an asymmetric reflection.

RH And you would know whether to anticipate relaxation based purely on the composition of layer you've tried to grow and the thickness of the layer?

BT You might expect relaxation in a layer that has a big mismatch. What do I mean by big is the next question you are going to ask me. If you are over something like a few thousand parts per million mismatch you have to watch carefully for relaxation.

RH Are there any conditions under which a symmetric geometry is not used?

BT Yes, it's when relaxation is serious. Though having said that many users set-up in standard X-ray geometry without thinking.

RH And they don't get the results they want?

BT They will be in error, but it depends on how they interpret it.

RH Well can you compensate, is there any mathematical way of doing that?

BT No because the X-ray experiment does not measure the relaxation, it simply measures the interplanar spacing, and it will give you a correct answer for the interplanar spacing. However, you've got a mixture of the strain associated with the fact that the composition of the top layer is different from the bottom, and the difference from what you expect it to be because of relaxation, so you have two parameters and there's no way you can distinguish those two unless you go to an asymmetric reflection.

RH Would you ever simulate just the substrate?

BT Yes, you would, and it is done quite regularly. An number of users use double axis X-ray diffraction to screen substrate quality, and so they will simulate it for comparison.

RH Is that to get a control condition or a standard?

BT Oh, you would probably use it as a standard, and we took great care in RADS to make sure that our substrate reflections agreed with everything else in the literature, because that'll be used by people as a test of whether the materials they bought from a substrate vendor are actually up to the specifications they require and to the specifications the vendors says they are. I know a couple of users who repeatedly screen with our instruments. That is not the main purpose of the instrument, but they do do it. Many people don't.

RH What happens if they don't (screen the substrates)

BT I don't know, I assume there units must be less, but maybe they're not.

RH Well I suppose you can control from a reputable vendor?

BT It's a question of quality control and probably not relevant to your study.

RH But it's nice to know if that's an important factor or not.

BT Yes, there's an important reason for simulating or looking at the substrate, although you may look at the substrate peak when you got the layer on, and that's to see how much curvature there is associated with the epitaxial layer, because in the real experiment the substrate peak will be broadened because of this curvature.

RH Over what range are the A:B layers of the MQW grown? So how many repeats are there?

BT P.Frewster at Philips research labs simulated a structure of 1051 layers; that I think is the world record. Typically I think people will go for 10-30 period repeat.

RH In the labs that tends to be the case?

BT It depends on the device. The problem is that very simply that device engineers are dreaming up all sorts of new structures.

RH So there are no rules for that?

BT I don't think you can build that one in there, as soon as you build it in you will find that someone has invented a new type of structure.

RH What are the likely ratios of the A:B layers?

BT (pause) They usually tend to be 1:1, 1:2 or 1:3, but people do grow a ratio of 1:10. I've seen an MQW with 18:180, 18:180.

RH Are there any typical compositions you can site for A:B layers?

BT People are trying to get as much variation with binary's because they are rather easier to control. You have fewer guns in your growth chamber. I think I've given you the sorts of things people are growing, but I wouldn't like that to be exhaustive.

RH Are there any impossible or improbable compositions?

BT I think you can assume that people are growing on 001 cube planes, and you can stick for the minute with cube orientation material. That limits you to a certain number of elements, and I would have thought that you either you would stay with Si and Ge or the III-V group elements Ga, In, As, P, Al to replace Ga, At, and also the II-VI in which you take Cd, Te, Cd and Zn, Se and ZnS. Oh yeh ZnS on Ge does exist as single layers, but I don't recall seeing ZnS Ge

super-lattice, but I would not be surprised if somebody hasn't tried to grow one.

RH Can grading occur through A:B layers?

BT Oh yes it can, particularly with the II-VI compounds CdHgTe and CdTe very serious convolutional gradient curves.

RH Is that intended?

BT No No, that's not intended.

RH And what does that result in?

BT Very poor satellite visibility.

RH A noisy rocking curve?

BT Yes, and even a rocking curve without any satellites in there. You would just see the zero order peaks and nothing else.

RH How do they go about resolving that?

BT There is no way from the X-ray data you can do any better than that.

RH It can't be simulated either I suppose?

BT Yes it can, RADS will certainly let you put in grading in a repeat structure like that.

RH But you would see nothing on the rocking curve if you didn't get it in the experiment?

BT (pause) Well a null result is a difficult thing to deal with yes, but it is something in terms of an expert system, that if you actually get no result you have to say well why is it. For example, I can remember going to UMIST and they ran a rocking curve of ZnS on Ge and there was a very nice sharp peak from the Ge substrate, and then an incredibly wide peak, about 4000 arc seconds in width, which hardly crept above the background and at first site there was a null result, yet it was just there if you looked closely, but that really meant that the epitaxy was incredibly poor and the ZnS has gone down with a whole range of orientations. So if you don't actually see the peaks your looking for the first thing to realise is that you may not have a very good epitaxy. That's the same argument that goes for MQW structures, that if you got interdiffusion of elements between the layers, then that's effectively saying you haven't got a good epitaxy. So a null result is important and isn't something we've talked about that much.

RH Does an MQW structure tend to be grown straight onto a substrate or not?

BT No there is always a buffer, it may be a buffer of the same material as the substrate, usually.

RH Are there any other structural things that are certain about the MQW structure.

BT You can usually guarantee a cap on the top, a thicker cap on the top.

RH How do the following affect the appearance of the rocking curve profile, and could you define each just to certain about what they are?

RH The spatial period of the structure?

BT The spatial period of the structure changes the separation of the satellites. The relative spacing of the two layers changes the relative intensity of successive satellites. In fact if you do a fourier transform you can see where that comes from.

RH The thickness of the repeating layer?

BT (pause) I didn't see the difference between that and the previous question. (pause) Oh the total thickness sorry. Well obviously the intensity goes up, the more layers you have the more intense is your signal, and you would expect to see small subsidiary interference effects associated with the total layer thickness, it would behave a bit like a single layer of that total thickness, but generally the more layers you have the chance you have of seeing it.

RH The composition of the layer?

BT The composition effects the position with respect to the substrate and the zero order peak and will affect the intensity of the satellites, and will affect the intensity of the zero order peak as well, but particularly the intensity of the satellites. So if you have something like AlGaAs and GaAs, then if there isn't much Al there then there isn't much difference between the scattering of the two layers then the satellite peaks will be incredibly weak. (RH So as the composition goes up you tend to see more satellite peaks) Yes, so if you have something like InP and InGaAs as the mixed MQW then you can get that lattice match so that the Bragg planes are quite close, because the In scatters from the Ga and the As and the P these two scatter really rather differently. The satellites are, therefore, very strong.

RH The dispersion of the repeating period?

BT Dispersion of layer thickness gives rise to broadening of the farthest out satellites first. Come back on that I'm not sure. So the higher order satellites begin to disappear in the noise.

RH Layer grading?

BT Layer grading results in asymmetry in the intensity of the plus and minus satellites.

RH Interface roughness?

BT This means instead of having an absolutely flat layer, you've got one that is sort of wavy. Now it's impossible to tell from X-rays whether you've got a wavy layer like that or whether you've actually got a grading of composition for an interface. So you see the same thing in the rocking curve.

RH Are there ever anymore than two layers in the repeating structure of an MQW?

BT To my knowledge I haven't seen one.

RH Do I actually need to think about it?

BT I think not. There is something called a Fibonacci sequence which is a mathematical sequence that doesn't give a regular period, but it does give peaks in rocking curves, and in a way is a sort of mathematical curiosity.

RH I assume if you have a very thin MQW structure there is a great deal of interference. Could you give an order in which effects should be included in simulation, for example, a what point would you include polarisation, the reference crystal, the relaxation percentage?

BT Yeh, if you'd done an experiment in which you'd got an unpolarised source, I think you should do the simulation in both polarisation states together. That may be a point of dispute though. I suspect KB would use sigma polarisation first, and suspect most people would use sigma polarisation first. Then you would include relaxation. At the moment most simulation programs don't include relaxation, RADS will do it for the symmetric geometry, but as yet won't do it for the asymmetric geometry, so that in a way would come last and you would put the reference crystal in second.

RH If layers get very thin this causes a shift in the Bragg peak away from where you expect to find it. Could you give some examples?

BT Now as the layer gets thin and the thickness at which this effect occurs depends on the mismatch. For high mismatch layers well its a percentage shift. This is a slightly tricky question to answer because it hasn't really been hacked out. For different mismatches there's a guy called Hui We over at Buffalo who has actually plotted this out for a whole load of structures, and what he finds is that for a 2% shift the product of the effective mismatch multiplied by the layer thickness is equal to 3.7 or maybe 2.7 I'm not sure, but it's a constant. That's independent of what the composition of the layer is, so clearly as you go further out the percentage shift gets bigger, and I just

have a feeling it might just be a thickness effect. Ball park figure, if your layers are less than 0.5 microns thick you need to worry. The shift is towards the substrate peak, the splitting is reduced. I don't understand the physics behind that, it's a physical effect. It's observed experimentally and comes out of all simulation programs, but I haven't managed to find anyone who understands why.

RH Last question, layers can appear on the rocking curve when they are not actually there. Where does that occur and how?

BT (Figure 3) On a GaAs substrate GaAlAs its .5 .5 so its 50:50, then then Ga .67 and Al .33 As and then on top you have Ga .5 and Al .5 As. Now if these layers are typically less than .5 of a micron each, say typically about .3 microns and then .1 microns in the middle there you'll get very complicated rocking curves from that. You will find that instead of having a substrate peak and then two layer peaks you will go from a substrate peak and then the layers split into four peaks or something like that. (see Figure 4). How you sort that lot out is not at all obvious, because you initially think you've got four compositions. In fact you've only got two compositions, and the only way to find out is to start with the structure you think you've grown and simulate the structure and then iterate. So this is why I emphasize that the input from what the grower thinks he's got is crucial.

Interview 4 Brian Tanner

RH From the first interview you classified rocking curves into eight different types: Substrate only; single layer; single layer with grading; multiple layer up to about six; multiple layers with grading; MQW structures; super-lattices with extra layers; and I have also included here possibly graded interfaces on a super-lattice. What I would like to do is produce a list a key features which distinguish each of these in terms of rocking curve profile only. Dealing with each in turn could you say what key features there are in the substrate that are important to its characterisation?

BT So you want to start off with just the rocking curve of the substrate?

RH Yes

BT One peak usually, but you have to be a little careful in that if you have set-up your experiment so that you have crystals that are not the same type in the double crystal geometry then you can see two lines sometimes associated with different lines in the X-ray spectrum.

RH Is this the reference crystal that is a different type to the substrate?

BT That's right, the reference crystal has not the same Bragg reflection. Then you can see two peaks, but they have a characteristic feature that they are usually in the ratio of 1:2, and you can calculate the separation that you would expect for any combination. So that comes from Braggs law. So you can easily check that out, but assuming that you've got the same crystal on the reference as the specimen, and the same Bragg reflection, then you will see one peak only.

RH Is there any reason why you would have a different reference crystal?

BT For convenience. There are beam conditioning monochromators, there's the four crystal monochromator that Phillips sell, which actually removes this possibility, but if you have an expert system it's something that you will have to put in the back of its mind, and how it does that I don't know.

RH And anything about the widths or any other features like that?

BT You can calculate what you expect the fundamental width to be for a perfect crystal.

RH Is that the half width?

BT Yes, and that you can calculate from dynamical theory, or using RADS.

RH But, you wouldn't necessarily use RADS in a simple substrate would you?

BT OH you might, because you might be interested in knowing if the substrate is significantly worse than it should be, because if there are defects there it will lead to a broadening of the rocking curve. So if you've got a width that is enormously large compared with what RADS would predict, you know that this is a very imperfect substrate. If it's too narrow you know you've got something wrong with the experiment.

RH Any other features you can think of for substrate that would be important (pause) height for example?

BT Height depends on the intensity of the X-ray source.

RH What about the wavelength of the X-ray source, how does that change the width of the peak, or the arc spread?

BT Well, the longer the wavelength the wider it will be. That again you have to calculate from RADS.

RH And the single layer?

BT The single layer you will be two peaks, if the layer is thick enough, and if the layer has a different lattice parameter to the substrate.

RH And what about the distance between the two?

BT Depends on what you grow. If you are growing GaAs on Si then they are a degree or so apart. If you are growing InGaAs on InP then there may be only tens of seconds between them. If you've got it absolutely right then it may be just a bump on the side of the substrate peak. I should say that in a symmetric geometry you would expect the substrate peak to be symmetric, and an asymmetry in the peak is a clue that you've got a layer present.

RH And the single layer with grading?

BT A single layer that is graded has a rocking curve that is broadened, and is usually sort of a wedge structure.

RH Does a change in the composition determine the direction of the wedge to the left or to the right. In other words, the increase in the composition of one element cause direction?

BT Yes.

RH And which way?

BT Pass. Hold on a moment, I think the lattice parameter (pause) Stop the tape. (BT not certain of answer) I think that if the lattice parameter at the surface is higher than at the interface with the substrate then the low

angle side is high compared with the high angle side. But I would need to sit down and actually check that out. I have not thought about that. I'm not sure if that's a rule that follows systematically all the way.

RH Would that follow for the material that's actually changed in composition?

BT No that's not right. I have an example here of a material that's -200ppm at the interface and -1000ppm at the surface, and we see that the left hand edge is higher than the right hand edge.

RH Could you draw a diagram of that?

BT (Figure 1) It's a linear grade from the surface - 1000ppm to -200ppm. This is the substrate and this is the surface, and that has a rocking curve that is higher on the low angle side. That is an decreasing lattice parameter going towards the surface. I recall most of them being with the wedge higher at the low angle side. I'm not even sure if that is a general criterion. I don't remember seeing any going the other way. As you see from these examples going through Martin Hills thesis, they are all wedges going ... are there's one going in reverse. (pause). If you take Figure 6.7 of Martin Hills Thesis you'll find that there is an example where he goes from -200ppm to -1050ppm at the interface and then reverses that grade so that it goes the other way, and the rocking curve does actually change its wedge angle for positive to negative. That would appear to give use the clue that for smaller mismatch at the surface its low low on the wedge side. I'm not sure, looking at Martin Hills thesis there appears to be some discrepancy there, what I will do is run some simulations on RADS for you just to check that out. That there is a pattern emerging.

RH Any other features for the single layer with grading?

BT You can work out roughly what mismatch the grading starts and ends from the end points of this rather broad wedge structure (marks on Figure 1), but it doesn't follow exactly. That gives you a good first start if you calculate what would be the composition corresponding to a layer of that mismatch. The good news is that many layers these days are not graded anything as much as that. The control over composition is much much better these days. It is rare to see a layer that badly out. In the industrial environment, if they got one like that, they you say it's graded throw it away. Having said that though there is a guy at Texas instruments who certainly was interested in that, because people also grow deliberately graded layers.

RH The next one is the multiple layer. You called that a class. Why did you actually separate that as a class (2-6 layers).

BT Yes, you tend to get here a situation where sometimes you can identify one peak for one layer, and sometimes you can't.

RH Does that mean that for certain layers the peaks may be missing from that type of structure, so there would be missing information and that would be significant, whereas you would not get that with other classes.

BT Well on a simple argument you might argue that for every composition of layer you would see one peak in the rocking curve, if you differentiate Braggs law that's what it will tell you. The problem starts to occur if you have two layers of equal composition sandwiching a fairly thin layer. Then interference effects can give you a situation where you may get that single peak splitting in to two, and all sorts of crazy things start appearing. So you can't in that situation immediately identify a peak composition and layer number per peak so you have to know the sort of structure that you've grown, and be prepared to simulate that from the start.

RH So the rules that would apply to that type of rocking curve would be slightly different. You would tend to look out for different things?

BT Yes, I would look for interference fringes to start with because on a thin layer structure you would see a significant amount of interference, and the first thing you would do is do physically or mentally a fourier transform to pull out the layer thickness from the periodicity. That would then give you a handle on certain of the layer thicknesses. From that you can then start simulating to try and fit the given composition for the peak position.

RH Anything else?

BT Now this is the most difficult of all possible worlds, where you have several layers separated by thin layers, then it does get extremely complicated. I'm not sure how I would go about it apart from knowing the model structure the grower has thought of growing, and going virtually straight away to simulation.

RH Would it be worth creating sub-classes for those types of structures where you have an inherent complexity in the rocking curve profile? So, for example, if you have very thin layers you would create that as another category?

BT (pause) I'm not sure if I know sufficient about the characteristics about a given situation to be able to do that.

RH O.K. what about the MQW?

BT The MQW or super-lattice is characterised by satellite reflections. So there are lots of little peaks which are equally spaced about a zero order peak which is usually displaced from the substrate peak. Usually, but not always

that is the highest peak in the structure, but not always. The satellite peak widths vary systematically in the structure. If you get significant broadening of the high order satellites then you are starting to find that there is dispersion in the layer thickness or grading.

RH What about the overall spread of an MQW?

BT The spread will be large, and that will depend on the individual layers concerned. The thinner the layers you grow the further apart the satellites will be. Again, a quick calculation, knowing what the grower thinks they've grown, it's again just differentiating Bragg's law, this will tell you what the satellite separation should be, and again that is something you would normally do as a starter in order to know how to take the data.

RH Setting the experiment up in other words?

BT Yeh, otherwise you don't really know over what range to scan.

RH O.K., and the super-lattice with the extra layers?

BT (pause) That changes peak intensities, but quite how, I don't think I've collected enough information to formulate general rules.

RH RADS would show that?

BT RADS will simulate it, that's not a problem, but I haven't systematically gone through looking these ...

RH It sounds as if I need to get a working copy of RADS to go through and try all these things out?

BT I think it's been agreed you should.

RH It's just that I didn't have a working version.

BT There are now working versions, and I think you can take a serial dongle with you if you ask nicely.

RH I will

BT There's even a manual on the shelf. You can't have a parallel dongle we haven't got enough of them.

RH It doesn't matter, I don't know the difference.

RH The graded super-lattice, that's the most complicated structure I assume you are likely to come across (BT that's correct), would that have the broadest feature range?

BT What you find is that the (pause) certainly you have to look at the higher order satellites and look for broadening of the satellites, and you also have to look for asymmetries in the satellite intensities about the zero order peak.

Until I started looking in some detail at RADS I was under the impression that for fairly low order of super-lattices you still have a nice symmetric profile. I've now come to the conclusion that when you only have 20 odd layers then the profiles are intrinsically not symmetric. So from what I said in the first interview I have to slightly backtrack on. Broadening of the high order satellites tells you that there is either dispersion of the layer thickness or grading at the interface. There are things that hold for a 500 period superlattice that doesn't hold for a low order period of about twenty layers. Unfortunately the twenty period ones are those that are incorporated into devices than growing for testing X-ray optics. So those are the ones that we have to get to grips with.

RH How do you go about running an experiment?

BT I walk into the lab and the student has a rocking curve on the screen and I've no idea what's there. My first thought is how many peaks are there that I can see. My second thought is how high are they in relation to one another. My third thought is where abouts are they with respect to one another. I know I can scale with the differentiation of Braggs law, and I know that roughly 4ppm corresponds to one second of angle for the 004 reflection, so I can immediately make a guess at the mismatch between layers if I have two layers there. I would then enquire, probably on a logarithmic scale, whether there are any fringes, and if there are any fringes then I would immediately say that that gives me a layer thickness. I don't know what layer it is at the moment, but at least one of them or a sum of layers has a thickness corresponding to a certain amount. The highs would also give me an indication as to whether that was consistent, and probably at that point I would hit the computer for a simulation. The peak shape is something that we sort of passed over, and I suspect it would come about now when I would iterate back and look at the peak shape and see if the rocking curve is asymmetric in which case I would suspect some broadening or another layer present, and also the wide of the layer peak to see if that was significantly broad. If I come in and see there are two peaks I would also want to check on a logarithmic scale whether there were any satellites along way out. I did exactly this with a sample that NL had from Bob Sacks and looking further out when low and behold another little one appeared (peak) when I said ahh this is a MQW isn't it Neil. So that was the clue that it was a MQW or a super-lattice, and then I would go and search an equivalent distance any the other side for the other peaks (other side of substrate).

RH Will you always find something on the other side if it's a MQW? Is that proof of it?

BT Well yes, it may not be outside the noise of your experiment, but yes in principle it's there. Then I would simply measure the separation of that satellite and that would give me the super-lattice period, the total period of

the whole thing, which means I've got layers A:B of total layer thickness $A + B$, and I would know the average composition from the other big zero order peak which is the composition of $A \times$ the layer thickness $A +$ composition $B +$ the layer thickness B divided by the total layer thickness - I think. Then you have straight out from direct measurement four variables, two directly. So that at least gives you a rather more limited set of things to try.

RH I assume when you solve a problem you can do it in two ways. Perhaps you can tell me the way you think you might do it. You can try to match the source or in this case a typical rocking curve description in a database of typical rocking curves to the target or experimental rocking curve by accounting for the quality of the sample, for example features like widened peaks, curvature, or strain, so that the prototype source rocking curves incorporate these dimensions as part of the overall classification. In other words you might include that when you are solving your problem. However, you might do it in another way, you might actually take your experimental rocking curve and eliminate in your head all the things that make-up the quality of the crystal and produce an idealised rocking curve, and know what it would look like if you didn't have peak widening, and if you didn't have curvature, and if you didn't have strain, and use that to actually try to find evidence to prove what the rocking curve is.

BT I think you do the later. You tend to look at substrate peak and it's broader than the theoretical value you will say there is curvature there and then ignore it. When you've actually got a fairly reasonable fit to the rocking curve structure you then iterate back and add a bit of curvature in. If you look at the way RADS is constructed that's the principle in that it's a convolution of ?? angle which you can add in on the graph plot afterwards.

RH So do you think expertise is the ability of the expert to imagine what the rocking curve would be without this extra dimension of quality. (four dimensions previously outlined of feature density, feature count, rocking curve type, quality). In other words you can eliminate that because once you know the other things (density and count) you can very readily classify the rocking curve.

BT I think the only one that really causes trouble is when the layer composition changes very significantly through the layer (Figure 1). If you have a small amount of grading through the layer or a diffuseness at the interface then what happens is that the peak shape doesn't quite fit. So you tend to say that there must be some grading there, and that's the ultimate fudge factor that allows you to say that I think I know what it is, but it doesn't quite fit. I think your second approach is the one we tend to use.

RH Are there any exceptional features that draw your attention for each rocking curve type? For example, if you were trying to identify a group membership of target DOG to

the prototype source of DOG-CLASSES and the dog had very large ears the you might focus attention on ears and that would be the tract to focus on when trying to prove class membership. For each of those classes you identified from substrate upwards can you say key features?

BT Yeh, the key feature with the substrate is only one peak. Single layer is two peaks, MQW is all these little satellites which are equally spaced. A big graded layer is this characteristic wedge shaped rocking curve. A very complicated structure with bumps and kinks on the peaks so there appears a modulation of a smooth intensity is a good clue to a multiple structure of moderately thin layers, and this is going to be the difficult one to handle.

RH With the negative version the same question, ie. features that are missing, for example, if the dog didn't have any hair you would very easily find out what the dog was by class membership. Applying that to rocking curves, how does that apply?

BT If you think you've got a single layer, but you can't see a peak then that probably indicates that the layer is either very thin or it's mismatched by more than you think it is in which case the peak is outside the scan range. If you think you've got a MQW and you can't see the satellites you come to the conclusion that the layers aren't very uniform or that there is a very significant interdiffusion of elements between the layers so that it not a really well defined square period, its some fuzzy amplitude modulation. So if you don't see satellites in the MQW structure you turn to the grower and say that this isn't very good is it? If you have a rocking curve which has just two peaks and the intensity is very much lower than you expect, this usually means that the rocking curve of the layer is broader than you expect, that really tells you you haven't got very good epitaxy, there are lots of defects at the interface, maybe lots of defects in the layer. If you don't see any peak at all then you haven't done a very good experiment, but I'm not sure if that's helpful. Certainly if the substrate peak is narrower than theoretically predicted the experiment is incorrect perhaps with a drift in the camera. So in a way that's a negative term.

RH I have produced a frame based system that calls external procedures when value and defaults are not applied (print-out of put-domain part of knowledge.lsp). For example, if the user knows neither the value, or is not sure if the default value of 004 is the correct reflection indicies for the experiment, then an if-needed facet is applied that automatically runs the procedure called bragg-law-satisfied. This runs through the basics of bragg law to help the user calculate the reflection indicies. Could you comment on the expression of differential expertise?

BT The importance in terms of the sub-routines (external procedures) is to calculate the angles between planes in the cubic system, say with cubic for the minute because most of

the materials in the electronics industry are cubic, because if you had an asymmetric reflection (pp 82-83 of RADS will show you exactly what that's doing) the angle of incidence of the X-ray beam with respect to the surface is different from the angle of the exit beam. In order to know what angle to put in with respect to the surface, which is what the user sees, then knowing how to calculate the Bragg angle, which is rather easy to do, but they also need to know the angle between the surface and the crystal planes they're interested in. That's something you can do very simply, but people have a habit of a mental block in actually doing it.

RH So in other words, this is where you would have your if-needed procedure which would call up a function that would help them interpret that (print-out of put-domain part of knowledge.lsp the reflection geometry slot).

BT That would tell you what would be the incident and exit beams by calculating the angle between the Bragg planes and the surface, and then giving you incident and exit beams. This helps you set-up. It would also be useful I suspect to say which direction you need to align your experiment. So that will be not so much for interpretation, but telling you how to start the experiment.

RH This particular frame is the target frame (experimental-rocking-curve) which is the set-up frame. In other words, that is exactly what that would do.

BT Fine

RH Discuss the asymmetric reflections for this frame?

BT If you have an asymmetric reflection you get a different peak splitting depending on whether you have a low angle incidence beam or a low angle exit beam. The splitting between the peaks is bigger if you have a low angle incident beam than for the exit, and this effect can be quite dramatic. The low angle exit, the layer peak is actually narrower than the low angle incident for the equivalent layer.

RH I assume what it is doing is going deeper or shallower into the sample.

BT No it isn't. The actual penetration is the same. It's actually all to do with the amount of X-ray beam accepted of the reference crystal, that's a bit subtle and I suspect irrelevant to the expert system. All you need to know is whether it does or doesn't. The graded layer (p191 of RADS)

gives a nice example of a linearly graded layer, and then a MQW with lots of satellites peaks. Notice on a linear scale you only see one little layer peak. That gives you the indication that on a MQW you've got to look at a rather low intensity. So the signal to noise ratio has to be really quite good.

Interview 5 Neil Loxley

RH (opening discussion) Yesterday we spoke about defining source prototypes in a dimensional space, we spoke about feature density, the total number of peaks in the rocking curve, and the type of rocking curve, arranged as a matrix. I would like to define an equation for finding the percentage area under the rocking curve or a specified arc range. In order to do this I would need to define a specific arc range and then do a percentage of that (Figure 1). Are there equations that would allow me to work that out.

NL Really the problem is that the peak shape is not well defined all the time, so you haven't got a nice curve that you can say that this is gaussian function so therefore work work out the area. What you would have to do is a numerical integration of the data to get the area underneath. The way KB defined the peak is if the point is three standard deviations from the peak the count rating would still have to be dropping for it to be significant, and then he finds the turning points as to where the peak ends.

RH So it would be possible by doing an integration under that curve to find that percentage area. I assume who ever wrote the software would know how to calculate the area?

NL KB has already included an integrated intensity measurement into RADS. So he must have worked out how to do it, so he's the one to ask.

RH I need to normalise that effect, and I was talking to BT yesterday and he said what he thinks the expert does is he takes away all the other effects when he's thinking about how to solve the problem. In other words, he takes away what grading and strain and layer thickness would do to the curve. He visually does that, and has an idealised version and compares that to what he thinks it would be. So in order to actually match this into a source prototype you need to take away the effects from the curve, and you need to normalise the area for things like the wavelength of the X-rays, which obviously shrink the rocking curve and make feature density rise, but you don't what to include that effect, you want to normalise things like peak shift which is due to grading and strain, and asymmetry, unevenness of layer thickness. Is there any other factors I would need to normalise? These are crystal quality factors.

NL Dislocations which sort of comes under strain a little bit. Curvature again strain. These are the two that spring to mind that aren't on the list.

RH Can equations be defined that take those effects away from the rocking curve?

NL The effects of dislocation is not very easy to model because you are trying to model all the effects of all the strain fields of all the dislocations there, and some of them are in different directions so the strain fields will

be quite complicated, and a lot of computation will be involved in trying to add all the effects of these individual strain fields. You can probably approximate it in other ways. You may find a relationship between dislocation density and peak broadening for example.

RH It would be ideal if you automated the procedure, you could I assume manually work out the area simply by measuring it and integrating under the actual plot, but that is not the way to create this matrix. You would want to do this automatically from the rocking curve that has been created from the DCC software, but I assume if I did that approach (automation of matrix definition) it would require a lot of working out to give you a proper indices for feature density. Do you think it's possible?

NL (pause) I think it's very difficult because of the unpredictability of what's involved. I guess you can get some information to go some of the way.

RH It doesn't have to be an exact percentage, but so long as it's within a certain range.

NL So what exactly are you trying to do, are you trying to get the integrated intensity?

RH What I'm trying to do is get the percentage area of the features of the rocking curve within a particular arc range over the total arc range of the rocking curve. So in other words, around the substrate peak, so what you have got in effect is a measure of the concentration of the information around a normalised point. If you are trying to analyse rocking curves and it's very difficult to pick out why an expert can categorise, and one of the effects was that feature density might be a very general concept which they use to classify rocking curves, because there are so many different types of rocking curves created and invented in the future, you've got to have a very generalisable system. So feature density might be a good way of defining of the axes or dimensions for defining rocking curves. That was the reason for it. (NL I understand) So feature density would be the Y-axis, the X-axis would be feature classification which BT has already given me, the other axis the Z-axis would be the total number of features which was the next question I was going to ask.

RH Would you define a procedure for counting the total number of peaks in a rocking curve automatically. Do you think that's possible?

NL Well it's already done in DCC, so it comes up with a peak list. I'll show you that.

RH I assume that's based on what's visually there. In other words, if there's one peak hidden in another then it won't pick it up?

NL Right it won't pick it up.

RH Say for example the peak looked like (Figure 2).

NL It might do.

RH So this is the DCC software with the automatic peak counter. Could you describe what's going on?

NL Right, what we are looking at is a curve with a strong substrate peak and some fringes, which I don't think are fringes, but satellite peaks from an MQW structure. We can't use the find peaks in logarithmic mode at the moment because it hasn't been written for it yet. So shift it back to linear mode.

RH Do you think that would be necessary to get a proper feature count.

NL No it's not because you are still using the same data, it's just when you are looking at it on the screen, it's easier to look at the smaller peaks in logarithmic mode because they are more dominate. I'll window that so we can see the smaller peaks.

RH So what you are saying is that it wouldn't come up with any different number with different modes?

NL No no it's just a display type thing. O.K. so we do find peaks the first thing it asks for is the number of standard deviations, there is a standard deviation test which basically means it finds all turning points in the curve. So every time the intensity turns it calls it a peak. Then it applies a significance test and says is this peak still going down at a certain number of standard deviations from the peak, and that's the number that you put in here. So if you put zero in it will find all the peaks. If you put a large number in it will only find string peaks. Three seems to be a good balance to find peaks that are really there. So we put three in. So first it finds 181 peaks but thinks only one of those is significant.

RH What significance level is it setting that at?

NL Three standard deviations.

RH Oh

NL I'll change that to one and it still finds 181 peaks, but now it finds a lot more that it thinks are significant.

RH No, I was wondering what the significance level was, was it 5%, 1% or .1%?

NL I don't know. KB will know.

RH KB will know.

NL So now it's found a lot more significant peaks. If I try two standard deviations we'll probably get a compromise between the two.

RH That figure will completely change depending on what you consider your significance level.

NL (pause) So it's found three peaks it thinks are significant. After it's found these peaks you can then go through and display them, saying "Is that a peak or not?" So I can choose one of these bumps and say that that's a peak, and it gives me peak intensity and half width. Now if we go into RADS and load a curve, and run a find peaks because the data isn't noisy, because it's been generated, unlike DCC experimental curves, there is no standard deviation test. So it finds all the peaks that are there, and you see it's doing an integrated intensity measure of all the peaks now. So it calls the strongest peak 100% and then it scales the others from that 100%.

RH So one of the things with the DCC software is to try and find the significant peaks?

NL Yeh, try and distinguish them from peaks that are caused by noise. A good test usually is that you can usually say a peaks significant after you've applied a smooth to it you can still see it as a peak.

RH What's the smooth?

NL Smooth is a three point smooth, it averages over three points. So it takes the three points adds them and divides by three effectively, and smooths out any sharp spikes.

RH And it does that successively for point after point after point.

NL Yes. I'll just load the previous curve and show you on the DCC software.

RH So it's like a running average. (pause) Why does that eliminate noise?

NL Because noise goes up and down very quickly, if you got one point that is higher than the next two then averaging over there brings that down considerably. Whereas if you have three points which are slowly climbing up the peak, when you smooth it the difference of averaging over the three points is quite small. Now you see that in logarithmic it's quite noisy down that bottom end, and in the tails you are not quite sure if there is anything significant there or not, but if you do a smooth you can see that a lot of the noise goes, and the bumps that have survived the smooth are probably quite significant. If you do another smooth you can see that some of those are surviving.

RH I see, so what you do is successive three point smooths. Is that how it works?

NL Yeh.

RH Where did that three point smooth come from?

NL KB, I think you need to interview him about things like this.

RH So after this smoothing it means that you get more significant peaks?

NL No the peak count only picks up a peak if it was there in the first place even though it's working on the smoothed data, all you are doing is removing some of the insignificant peaks.

RH So it doesn't actually increase your confidence, although it should in theory if the three point smooth is working properly because your confidence in the surviving peaks goes up.

NL We'll try it. I've never really tried this exercise before. I'll run a couple of smooths ... before it only found 1 significant peak, but now it's found quite a few significant ones.

RH So it's increased confidence levels

NL yeh

RH Does that go up the more you smooth?

NL Don't know. Let's try. Do another couple ... no. What is happening now is that we are actually losing peaks.

RH So I wonder how much you know how to smooth?

NL Well one or two three point smooths will get rid of most of the noise.

RH How come that's not done automatically?

NL What smoothing? (RH yeh) Well you don't really want to smooth data automatically, it's not very scientific, you should keep data as true data, and really you should analyse the real data, but it's a useful check to get rid of some of the noise if it is noise.

RH Could you run through how you perform the experiment?

NL Lets go to the lab.

RH (In the lab) Say a few words on the setting up of the instrumentation?

NL O.K. what I've already done is aligned the instrument so that we are getting a significant intensity through the collimator, and I've already adjusted the first crystal so we have a strong Bragg reflection from the first crystal.

RH This is the reference crystal?

NL Yes. Know I've arranged the crystal so that it passes over the second axis at the correct height and directly over the centre of the axis. Now that's important because if you don't do that you get wider Bragg peaks. What I'm going to check now is that the beam is where I think it is and that I've still got the intensity there.

In the next stage is to check the first crystal, turn the counter on the DCC and open the X-ray shutter, and you see I have a nice strong intensity coming from the first crystal. Now what I want to check is that the position of that beam is correct so what I do is put a couple of alignment tools in and make sure that the beam can pass through both the horizontal slot at the right height. If the beam can pass through that slot then the beam is at the correct height. At the DCC screen there is still significant intensity coming through. However, it's not as strong as it should be, so what I can do is change the tilt of the first axis which has the effect of moving the beam up and down.

RH So what you are doing is counting the X-rays via the received through the detector, and the more there are the better the alignment.

NL Right, because we are interrupting the beam with the slot. So what I'm going to do now is do a scan on the first tilt goniometer, which has the effect of changing the tilt, and effectively moves the beam up and down. So you should get a very strong intensity when it's passing through the intensity when it's passing straight through the centre of the slot. When we stated it was only 8,000 and it's up to 18,000 at the moment. Now it's dropping off so I'll go back a stage. I'll do another scan, but in the other direction.

RH What does C mean - complete scan?

NL C means centred, which means it does the total scan range that you put in except it first winds back half the scan range and then goes plus from that point. So it does the whole scan in the same direction which is important for consistency and getting rid of backlash, but it arranges the scan so that the centre of the scan is where you start effectively.

Now you should start to see the count drop off quite quickly when it reaches the top and bottom of the slot.

RH This suggests that -3.5 is the correct position for the tilt?

NL Yeh, so as soon as it hits the edge of the slot it starts to drop quite quickly. So I will stop it there and move the goniometer to -3.5.

Now I'll take it back to count and put the other tool in which is a vertical slot used to measure the beams over the second axis.

RH You are moving that by hand.

NL Yes a bit naughty, but the X-ray scatter is rather small. Now when we put the second tool in it's cutting the beam which is bad, and I'm not really sure why.

RH Why are you getting counts of 0 and the occasional 17?

NL That is because we are using a gate time of 0.06 seconds, which means that if it happens to see a pulse in that 0.06 gate it scales it up to counts per second and 1 over 0.06 is roughly 17.

I've just checked to see if the beam is where I want it to be, now what I'm going to do is check that the theta rotation of the first crystal is still correct because when you change things they often change the angle of incidence of the beam slightly. So I'm doing another scan to make sure I'm maximising the intensity from the first crystal.

I'm looking at the curve now to make sure that there are some X-rays, and you see we are a long way off from where we first started. We are actually looking at the $K\alpha_2$ peak, and you can see that there is a slight bump there. That is more to do with the $K\alpha_2$ peak and the $K\alpha_1$ peak is slightly more to the left. So because we get more intensity from that I'll move the axis back to beyond the $K\alpha_1$. I'll do find peaks just to find the highest peak.

RH Would somebody operating the DCC know about the $K\alpha_1$ and $K\alpha_2$ radiation differences? Would they suspect that that fall off is due to $K\alpha_2$ and not $K\alpha_1$?

NL Well you tend to diffract both of them simultaneously in double crystal diffraction. It is possible to resolve them through careful use of slits, but if you just find the maximum intensity that's all you are interested in.

RH So it's highly unlikely that you would pick the wrong peak?

NL That's right, you are just looking for maximum intensity. The maximum intensity for this experiment is about 100 arc seconds from where we were.

RH What's the figure on the bottom right of the screen?

NL That's the full scale of the bar graph at the moment. So 64,000 count would saturate the display.

Now I'm happy about the first reflection, what I do now is mount the sample holder, and the first thing I have to do is check that the face of the crystal is intersecting the beam. So I actually put the sample parallel to the beam and I have to find a point at which that is cutting the beam. So I count again, measure the intensity, slide the sample forwards until I find it's cut the beam, and I find a point

at which it's about half the intensity, and I call that the point at which it's cutting the beam.

RH Why half the intensity?

NL Just a rough guide to cutting down the middle. It's not totally accurate it serves to within half a millimeter maybe. Now the crystal is over the axis, so I move the detector to 2θ which is twice the bragg angle of the reflection you are interested in.

When in position you increase the X-ray intensity. I was previously using only 18KV now to excite the characteristic lines from Cu you have to energise the thing to about 20KV.

RH The higher the intensity the more the peaks rise above the background?

NL No, once they are excited the thing goes linearly, but there's a point when you are only just starting to excite the $K\alpha_1$. X-ray intensity goes roughly linearly with milliamps, but not linearly with kilavolts. It goes up quite strongly as the lines are excited, but tails off again because the contribution from the characteristic lines in relation to the rest of it decline a bit, so it tails off with kilavolts.

Now I manually rotate the sample round to θ , and should be able to pick a peak up from there. Now the peak is very narrow so you are not going to get it very well adjusted by hand. If the sample is not orientated very well it's the quickest way to get somewhere close. I lock the sample holder into position.

RH Is this a symmetric reflection?

NL 004 symmetric reflection is the easiest.

Now I move axis two until I find the peak, so I move by maybe 1000 arc secs and stop it when I see a peak coming up. I'm not sure in which direction it's going to be in.

RH Is there a situation where you wouldn't know what the sample is?

NL You would normally know what it is, but the substrate may be one or two degrees off which is when you have problems when you try to find the peak with the fine axis because it takes longer to cover a larger distance. So I find the peak, turn up the intensity a little bit, do a quick scan over about 200 arc seconds with a large step size of about 4 arc secs, and 0.5 counting time just to see what's there.

Now I know what to expect because it's a single substrate, it should be just a single peak that should be quite narrow. What I then have to do is adjust the tilt of the sample so that the diffraction vector is lying in the plane of incidence. If it's tilted then the diffraction vector can lie outside the plane of incidence and you actually get some dispersion effects. You scan again to check that. Now you have a peak that is quite sharp and has a flat top which is due to the step size that we used. What I want to do now is

get a measure of the half-width which will say how well orientated the crystal is, because I know what to expect for this type of crystal.

RH What type of crystal is it?

NL It's InP. Now the half-width it's come up with is 16, but that will be ± 8 effectively because of the step size. So what I do now is move axis two through the peak and see what the peak intensity is. So you see a peak intensity of about 40,000. The effect of tilt is that it squashes and broadens the peaks, but the integrated intensity stays the same. What you are therefore doing is minimising the half-width by maximising the intensity. So what I do is move gonead 2 which is the second tilt axis by a certain amount. I'll move it by 1mm in the negative direction. Then I'll go back to axis two and see if the peak intensity has gone up or down. What was it at before?

RH It was at about 40.

NL Its know at 38 so it's gone down. That means I've gone in the wrong direction. So I'll move gonead 2 2mm in the other direction. Do the same with again with axis two.

RH Is this always done with the substrate peak only irrespective of the crystal type you are using?

NL Yeh, you would normally try and optimise on the substrate peak only.

The count rates gone up a bit so I'll move further in that direction. It's still going up a little bit. Move another millimeter. I'll go another millimeter, but it should start to drop a bit then.

RH This is the sort of thing that could be automated.

NL Yes some of it could. There are automatic systems that people have designed, but they tend to take a bit longer compared to those with user intervention when they can stop what's going on.

Still going up a bit so I'll carry on. Up to about 43 now.

RH So how accurate to you go in the second axis?

NL About 0.5mm I try and get it to.

Put the thing on the peak, but do a higher resolution scan with a longer counting time. I'll do a 1.5 sec step and count for about 2 seconds per point which should give me pretty good statistics.

RH When you say you know what to expect and that automatic machines tend to be a lot slower, what is it in the experimental situation that you expect? (NL not sure) You know what to expect and you are driven by your knowledge of rocking curves, could you articulate that?

NL The point is that you can stop trends much quicker when you are watching it than you could really statistically test for. I think that that is the key. You have to apply some fairly rigorous statistical tests so that the software would know when things have gone wrong. For example, it could gone on looking for a peak when you know intuitively that you've gone too far in a particular direction, but the stats system would still keep expecting it, or it would have to apply a statistical test to account for all situations, and that will have to take a certain amount of time. So you can do things quicker because it's easier to stop trends.

RH Anticipation is the key there. How long do these automatic systems take to run then?

NL No ones written one for this software. Not a long time, but there are not as quick as doing it by hand. Maybe two or three times.

RH I assume that if you are doing a lot of experiments in a day you want to do things quickly. Do you think there is a possibility that the processing power will overtake the manual method.

NL You are still limited by the time it takes to move motors.

RH So what you are saying is that automatic statistical programs, because speed is the result of the number of decisions that have to be made, the slowness is the fact a statistical program would have to make smaller changes in the axis, whereas a person could make bigger jumps.

NL You tend to use very short counting times when you are doing things by eye because you can see trends very quickly. Statistical tests would have to amass a significant count rate, which would mean counting for longer. It would have to make a decision as to what level to call things significant and to do that it has to decide how long to count for, and it's always going to be longer than a human operator would. That's the limiting thing, not the computational speed, because the number of decisions is quite small really.

NL I've got a rocking curve, and the first thing I notice about it is that it is very symmetrical which is a very good sign. Lack of strain. The background comes down quite quickly, and generally it's a nice shape. Do a find peaks, it has a half-width of 11 arc seconds which is quite good for InP. The best for InP is 9.5 arc seconds. So this result means that either there is a little strain there or the surface of the crystal isn't as clean as it could be, or there are defects in the crystal, but I would pass InP at 11 arc seconds as being a good crystal.

NL (Discussing experiment) This is the set-up of the experiment (Figure 2).

Appendix 2

Design Model used for Experimental K.E. of X-ray Rocking Curve Design Framework I

This appendix details the experimental design used for the experimental knowledge elicitation technique. (see Chapter 6: Section 6.3). There are eight experiments (E1-8) in experimental framework I, and these are summarised in Table I. In each experiment there are two sessions, a training session and either a recall or categorisation session. The training session consists of displaying batches of five distortions of a series of graphical prototypes from a selected domain (Even Functions, X-ray Rocking Curves) for ten seconds each. The distortions of four unique prototypes are used in each experiment, making a total of twenty distortions per experiment. These are displayed in prototype order using a series of A4 plots taken from a total population of 400 distortions. There are five subjects in the expert group, two subjects in the journeyman group, and six subjects in the novice group. During training, subjects are asked to observe the features of each curve and make a mental note of any observed characteristics. There are eight training sessions in the framework, one for each experimental type (rocking curve random, rocking curve rule, even-function random, even-function rule) presented first for recall and then for categorisation. In all cases each training set uses a unique set of prototypes, and the order of presentation randomised to prevent learning effects across experimental set-ups.

Interceding each training session is either a recall session or a categorisation session that uses data based on the training, resulting in the four training recall (Rc) experiments and four categorisation (Gr) experiments. Each recall session involves the re-presentation of three of the original distortions (Do) of each prototype from the training session together with three new distortions (Dw) of each prototype and the four prototypes (Pn) used to generate the distortions for the experiment. Subjects are allowed five seconds per plot to record whether or not they have seen the patterns displayed in the preceding training session. Experimentally, subjects only see twelve of the twenty plots shown in training a second time. Performance is a measure of how accurately subjects recall the patterns without direct comparison between plots. Performance is measured across subject groups and experimental type. Each categorisation session presents material in exactly the same configuration as the recall sessions, but this time subjects are required to sort the plots into four categories within a set period of time. Performance is measured by the accuracy of the classifications. Because cross comparisons are allowed between plots subjects can directly compare features. Table II summarises the data presentations in experimental framework I. The data presentations are

performed first for recall and then repeated for categorisation.

Table I
The design for experimental framework I

Domain	Rocking Curves				Even Functions			
	Rule		Random		Rule		Random	
Session	Rc	Gr	Rc	Gr	Rc	Gr	Rc	Gr
Expert	E1	E2	E3	E4	E5	E6	E7	E8
Journeyman	E1	E2	E3	E4	E5	E6	E7	E8
Novice	E1	E2	E3	E4	E5	E6	E7	E8

Table II
Data presentations for framework 1 using subject recall and category performance

Session		Domain													
		Even-function				Rocking Curve									
		Transformations													
		Rules		Random		Rules		Random							
TS	E -	20	X	Do	20	X	Do	20	X	Do	20	X	Do		
	J -	20	X	Do	20	X	Do	20	X	Do	20	X	Do		
	N -	20	X	Do	20	X	Do	20	X	Do	20	X	Do		
ES		Pn	4	X	Pn	4	X	Pn	4	X	Pn	4	X	Pn	
	E	Do	12	X	Do	12	X	Do	12	X	Do	12	X	Do	
		Dw	12	X	Dw	12	X	Dw	12	X	Dw	12	X	Dw	
		Pn	4	X	Pn	4	X	Pn	4	X	Pn	4	X	Pn	
	J	Do	12	X	Do	12	X	Do	12	X	Do	12	X	Do	
		Dw	12	X	Dw	12	X	Dw	12	X	Dw	12	X	Dw	
		Pn	4	X	Pn	4	X	Pn	4	X	Pn	4	X	Pn	
	N	Do	12	X	Do	12	X	Do	12	X	Do	12	X	Do	
		Dw	12	X	Dw	12	X	Dw	12	X	Dw	12	X	Dw	
		Pn	4	X	Pn	4	X	Pn	4	X	Pn	4	X	Pn	
		J	Do	12	X	Do	12	X	Do	12	X	Do	12	X	Do
		Dw	12	X	Dw	12	X	Dw	12	X	Dw	12	X	Dw	

TS=training session, ES=experimental session
E=Expert Group, J=Journeyman Group, N=Novice Group
Do=seen distortion, Dw=unseen distortion, Pn=prototype

The results are separately analysed for two performance factors, recall and categorisation, using an analysis of variance (ANOVA) in the form of a four way independent model with a 5% level of significance set for all factors. ANOVA isolates the variation of results across the design for subject groups (experts journeymen and novices), transformation types (rule or random), plot types (Pn Dw Do), and domain (control or rocking curve). The first Null hypothesis (Ho(1)) is that there is no significant difference in recall performance between the three subject groups. The second Null hypothesis (Ho(2)) is that there is no significant difference in the recall rates for plots of different types (Do Dw Pn). The third null hypothesis (Ho(3)) is that there is no significant difference in recall performance of subject groups for random and rule based transformations. The fourth null hypothesis (Ho(4)) is that there is no significant difference in categorisation performance across subject groups. The fifth null hypothesis (Ho(5)) is that the correct categorisation is invariant for plot types (Do Dw Pn). The last null hypothesis (Ho(6)) is that categorisation does not vary significantly against rule and random transforms. The results of an analysis of variance for the four main factors: domain, transformation, recall type and expertise are given in Tables III and IV.

Table III
ANOVA for recall rates in experimental framework 1.

SV	DF	SS	MS	F	S
Total	72	317.62			
Mean	1	237.22			
Expertise (E)	2	7.27	3.64	4.18	2.5%
Recall Type (R)	2	4.91	2.46	2.83	ns
Domain (D)	1	0.22	0.22	0.25	ns
Transformation (T)	1	1.20	1.21	1.39	ns
R x T	2	6.40	3.20	3.68	5.0%
R x D x T	2	6.34	3.17	3.64	5.0%
Other Interactions	25	22.8	0.91	1.05	ns
Error	36	31.2	0.87		

SV=source of variation, DF=degrees of freedom, SS=sums of squares, MS=mean square, F=F value, S=level of significance, ns=not significant.

Table IV
ANOVA for categorisation rates in experimental framework I.

SV	DF	SS	MS	F	S
Total	72	471817			
Mean	1	441330			
Expertise (E)	2	1513	757	3.00	5.0%
Category Type (C)	2	2941	1420	5.64	1.0%
Domain (D)	1	8866	8866	35.2	0.1%
Transformation (T)	1	2233	2233	8.87	1.0%
E x D	2	1747	874	3.46	5.0%
E x T	1	1892	1892	7.51	1.0%
Other Interactions	26	2329	89.6	0.36	ns
Error	36	9066	251.8		

SV=source of variation, DF=degrees of freedom, SS=sums of squares, MS=mean square, F=F value, S=level of significance, ns=not significant.

In both the categorisation and recall sessions, expertise varies within the X-ray rocking curve analysis domain. The even function domain acts as a control. For recalling data, expertise is a significant factor (2.5% level) in the subject's performance and (Ho(1)) can, therefore, be rejected at this level. This effect does not, however, extend to the type of domain since there is no significant interactive effects (D x E). Expertise is also a significant factor (5% level) in categorising data. This effect extends at the same significance level to interactions between subject group and the domain (E x D). It is, therefore, possible to reject (Ho(4)) and accept the hypothesis that expertise affects performance across domains.

If prototype structures operate in any of the experimental conditions then it is expected that subjects will recall Pn plots at a greater than expected rate when compared to the recall of Do transformations shown in the same session. Analysis of the data demonstrates that the (Ho(2)) cannot be rejected. This means that the types of plot presented does not seem to affect the performance of subject groups in the experiment. However, there is a fairly significant interaction (5% level) between plot types and the type of transformation (R x T), and this is extended to interaction between plot types, transformation types and the domain (R x T x D) for recalling data. For categorising data under the framework, analysis demonstrates that transform type is a significant factor with a rejection level of 1%. This means that (Ho(5)) can be rejected.

Each domain is sub-divided into rule and random transformations of data. With the random condition of both domains, subjects cannot infer structure following training and recall, they can only recall plots either through image

recall or prototype formation or both. In the rule condition subjects are able to infer structure provided they know the rules of transformation. When recalling data the results show that $(H_0(3))$ cannot be rejected since there is no significant interactive effect between expertise and transformation type. However, when categorising data, subjects are able to group plots either through feature analysis, prototype formation, or rule inference. In this measure of performance, the effects of transformation type are fairly significant (1% level), and this is reflected in subject group performance ($E \times T$) with significance levels of 1%. $(H_0(6))$ can, thereby, be rejected and it can be assumed that expertise plays a role in categorisation performance between rule and random transformations. Tables V and VI indicate that experts are more likely to mistakenly record the prototype than novices. Unfortunately, there is no significant four way interaction between expertise, categorisation type, domain and transformation ($E \times C \times D \times T$). However, experimental effects can be observed if the probability of recalling prototypes by chance is analysed. Results show that the probability of subjects recalling the prototypes due to chance across all experimental set-ups are in the range of 9%-20%. This is significantly low enough to indicate that experimental effects are operating in the framework. The most significant result is that of domain classification performance with a level of rejection set to 0.1%. This indicates that categorisation performance is markedly affected by the domain, with rocking curves being considerably easier to classify than even functions. This clarifies the common sense belief that expert are expert at analysing X-ray Rocking Curves.

Table V
Probability of prototype recall against expertise:

Data Set	Level of Expertise		
	Novice	Journeyman	Expert
DkTu	0.135	0.67	0.75
DkTa	0.145	0.52	0.39
DeTu	0.256	0.45	0.31
DeTa	0.12	0.34	0.31

De=Domain Even, Dk=Domain X-ray RC, Ta=Random Transformation, Tu=Rule-based Transformation

Table VI
Probability of correct prototype categorisation
against expertise.

Data Set	Level of Expertise		
	Novice	Journeyman	Expert
DkTu	0.815	0.755	0.98
DkTa	0.625	0.975	0.625
DeTu	0.4	0.321	0.45
DeTa	0.56	0.54	0.67

De=Domain Even, Dk=Domain X-ray RC, Ta=Random
Transformation, Tu=Rule-based Transformation

Appendix 3

Design Model used for Experimental K.E. of X-ray Rocking Curve Design Framework II

To test for the deployment of features by experts in the formation of prototypes, the performance of two subject groups, experts and novices, are compared when recalling data following a training session. Five experts and five novices make-up each subject group. The session format of Framework I is used again for Framework II, but with the data presented during training grouped according to selected features. Four experiments were conducted using the even spread of distortions taken from a total data population of 400 generated from 10 prototypes. The data consists of a series of graphical A4 plots on which a single rocking curve is displayed, and each of the distortions is classified into one of two categories, constant or variant, for each of the four features. The constant category consists of data which is organised to hold the selected feature experimentally constant with all other features variant. In the variant condition the feature is not held constant and has no visual pattern (see Chapter 6: Table 6.1). Within training Group 1, peak density is constant and, thereby, experimentally accessible. The other features, are not held constant and are, therefore, not accessible (see Chapter 6: Table 6.1). Groups 2, 3, and 4 hold respectively Peak Count, Peak Type, and Peak Position constant with all other features variant. The experimental framework compares subject's recall performance for different data types (Pn Do Dw) across the four training group classification scheme outlined in Table II against expertise (see Table VII).

Four experiments were each divided into two sessions. In the first training session subjects were shown 20 distortions of four rocking curve prototypes for 10 seconds each in prototype sequence. Subjects were unaware of the classification scheme used during training. This was repeated for all four classifications interceded by recall sessions in which subjects were shown the 4 prototypes (Pn) used to generate distortions in the preceding training session, 12 of the distortions shown in training (Do), 3 for each prototype, and 12 distortions not shown in training, but belonging to the same classification set (Dw), again 3 for each prototype, making a total of twenty eight data items. Each data item in the recall session is shown for 5 seconds in which time subjects either indicate that the item was shown in training or not. Performance is a measure of how accurately subjects recall distortions.

Table VII
Data presentation for framework II using subject recall
performance

Rocking Curve Training Sets

		Group 1	Group 2	Group 3	Group 4
TS	E -	20 X Do	20 X Do	20 X Do	20 X Do
	N -	20 X Do	20 X Do	20 X Do	20 X Do
ES					
	Pn	4 X Pn	4 X Pn	4 X Pn	4 X Pn
	E Do	12 X Do	12 X Do	12 X Do	12 X Do
	Dw	12 X Dw	12 X Dw	12 X Dw	12 X Dw
	Pn	4 X Pn	4 X Pn	4 X Pn	4 X Pn
	N Do	12 X Do	12 X Do	12 X Do	12 X Do
	Dw	12 X Dw	12 X Dw	12 X Dw	12 X Dw

E=Expert Group, N=Novice Group
TS=training session, ES=experimental session
Do=seen distortion, Dw=unseen distortion, Pn=prototype

Results are interpreted using ANOVA in the form of a three way independent model with a 5% level of significance set for all factors. The statistical design isolates variations due to subject groups (expert or novice), plot types (Pn Dw Do), and training group (1 2 3 4).

The first Null hypothesis (Ho(1)) is that there is no significant difference in recall performance between the two subject groups. The second Null hypothesis (Ho(2)) is that there is no significant difference in the recall rates for plots of different types (Do, Dw, Pn). The third null hypothesis (Ho(3)) is that there is no significant interaction between the two subject groups and plot types. Table VIII gives the results of ANOVA for the three main factors of expertise, transformation and group plus all significant interactions.

Table IX shows the probability of subjects recalling each type of plot (Pn Do Dw) following training with each of the four feature groups. The probability of subjects mistakenly recalling prototypes for each of the feature groups is shown against expertise in Table X.

Table VIII
ANOVA for recall rates in experimental Framework II.

SV	DF	SS	MS	F	S
Total	120	822			
Mean	1	731			
Expertise (E)	1	3.72	3.72	5.47	2.5%
Transformation (T)	2	4.21	2.11	3.10	5.0%
Group (G)	3	1.40	0.47	0.69	ns
E x T	2	4.56	2.28	3.35	5.0%
E x T x G	6	9.47	1.58	2.32	5.0%
Other Interactions	9	2.04	0.23	0.34	ns
Error	97	65.6	0.68		

SV=source of variation, DF=degrees of freedom, SS=sums of squares, MS=mean square, F=F value, S=level of significance, ns=not significant.

Table XI
Probability of recall each plot type
against feature type.

Group	Plot Type		
	Pn	Do	Dn
1	0.68	0.57	0.52
2	0.60	0.51	0.57
3	0.75	0.56	0.52
4	0.47	0.59	0.58

Recall performance varies across subject groups for the X-ray rocking domain. (Ho(1)) can be rejected on the basis of a 2.5% significance level. (Ho(2)) can also be rejected at a 5% significance indicating that plot type affects recall probability. More significantly, this effect translates to the (E x T) condition at a 5% significance level. This leads to the rejection of (Ho(3)) and the acceptance that subject group performance differs for different plot types. There is also a significant interaction for the (E x T x G). However, this is not an important interaction because the design attaches no significance to performance differences across feature groups. The probability for subjects recalling each

of the plot types (Pn Do Dw) is given in Table IX. The probability of recalling prototypes is the highest followed by distortions shown in training, and then unseen distortions of the same prototype sets. This significant result suggests that prototype structures may exist in the heads of subjects, that they are quickly formed during training, and that they are used during the recall session as a memory aid. Table X gives a more revealing interpretation of the (E x T) interactions, and shows that the probability of subject groups recalling prototypes varies against expertise. In groups 1, 2, and 3 the expert is more likely to recall the prototype in the recall session as compared to the novice. This isolates the effect of the prototype differences and indicates that experts are the ones who use prototype structures, fitting the training data to existing cognitive structure and consequently making more mistakes than the less knowledgeable counter-part when recalling data. The prototype effect exists for peak density, peak count, and rocking curve type, but not for peak position.

Table X
Probability of prototype recall
against expertise.

Group	Expertise	
	Novice	Expert
1	0.65	0.76
2	0.62	0.68
3	0.62	0.78
4	0.51	0.43

Appendix 4

Shows the consultation for a typical MQW structure
This is the historical record of the
expert system's operation

CONSULTATION HAS PROGRESSED

CONSULTATION BEGINS

Initial analog probabilities are:

Target analog = .4
Source analog = .3
Map analog = .2
Evaluate analog = .1

MQW frame uninstantiated

FIND NEW TASKS FOR AGENDA

TYPE-OF-STRUCTURE LATTICE-PARAMETERS LAB-SET-UP
tasks found for TARGET analog.

TYPE-OF-STRUCTURE task placed on agenda with priority of
.375

LATTICE-PARAMETERS task placed on agenda with priority of
.35

LAB-SET-UP task placed on agenda with priority of .325

BEGIN FORWARD CHAINING ON TARGET

Value MQW added to frame ALL-ROCKING-CURVES for TYPE-OF-
STRUCTURE task

Begin interrupt check for goal ALL-ROCKING-CURVES-TYPE-OF-
STRUCTURE=MQW

No demons found for task ALL-ROCKING-CURVES-TYPE-OF-
STRUCTURE

Constrained question posed to user for ALL-ROCKING-CURVES
LATTICE-PARAMETERS

User responses = 1

Apply constraints: ALL-ROCKING-CURVES LATTICE-PARAMETERS =
-PEAK-SPLITTING

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=NOT PEAK-SPLITTING

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

Apply constraints: ALL-ROCKING-CURVES LATTICE-PARAMETERS =
-COMPOSITION

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=NOT COMPOSITION

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

Apply constraints: ALL-ROCKING-CURVES LATTICE-PARAMETERS
= -MISORIENTATION

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=NOT MISORIENTATION

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

Apply constraints: ALL-ROCKING-CURVES LATTICE-PARAMETERS =
-TILT

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=NOT TILT

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

Apply constraints: ALL-ROCKING-CURVES LATTICE-PARAMETERS =
-RELAXATION

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=NOT RELAXATION

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

Apply constraints: ALL-ROCKING-CURVES LATTICE-PARAMETERS =
-THICKNESS

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=NOT THICKNESS

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

Value MISMATCH added to frame ALL-ROCKING-CURVES for
LATTICE-PARAMETERS task

Begin interrupt check for goal ALL-ROCKING-CURVES-LATTICE-
PARAMETERS=MISMATCH

No demons found for task ALL-ROCKING-CURVES-LATTICE-
PARAMETERS

FIND NEW TASKS FOR AGENDA

WAVELENGTH GEOMETRY REFLECTION-INDICIES STRUCTURE MILLER-
INDICIES ARC-RANGE STEPS SCAN REFERENCE-CRYSTAL tasks found
for TARGET analog.

WAVELENGTH task placed on agenda with priority of .39
GEOMETRY task placed on agenda with priority of .38
REFLECTION-INDICIES task placed on agenda with priority of
.37
STRUCTURE task placed on agenda with priority of .36
MILLER-INDICIES task placed on agenda with priority of .35
ARC-RANGE task placed on agenda with priority of .34
STEPS task placed on agenda with priority of .33
SCAN task placed on agenda with priority of .32
REFERENCE-CRYSTAL task placed on agenda with priority of .31

BEGIN FORWARD CHAINING ON TARGET

Default question posed to user for LAB-SET-UP WAVELENGTH
with default Cu

User responses = Y

Apply constraint: LAB-SET-UP WAVELENGTH = -SYNCHROTRON

Begin interrupt check for goal LAB-SET-UP-WAVELENGTH=NOT
SYNC
HROTRON

No demons found for task LAB-SET-UP-WAVELENGTH

Apply constraint: LAB-SET-UP WAVELENGTH = -Mo

Begin interrupt check for goal LAB-SET-UP-WAVELENGTH=NOT Mo

No demons found for task LAB-SET-UP-WAVELENGTH

Apply constraint: LAB-SET-UP WAVELENGTH = -Fe

Begin interrupt check for goal LAB-SET-UP-WAVELENGTH=NOT Fe

No demons found for task LAB-SET-UP-WAVELENGTH

Apply constraint: LAB-SET-UP WAVELENGTH = -Ag

Begin interrupt check for goal LAB-SET-UP-WAVELENGTH=NOT Ag

No demons found for task LAB-SET-UP-WAVELENGTH
 Apply constraint: LAB-SET-UP WAVELENGTH = -Cr
 Begin interrupt check for goal LAB-SET-UP-WAVELENGTH=NOT Cr
 No demons found for task LAB-SET-UP-WAVELENGTH
 Value Cu added to frame LAB-SET-UP for WAVELENGTH task
 Begin interrupt check for goal LAB-SET-UP-WAVELENGTH=Cu
 No demons found for task LAB-SET-UP-WAVELENGTH
 Default question posed to user for LAB-SET-UP GEOMETRY with
 default SYMMETRIC
 User responses = N
 Begin interrupt check for goal LAB-SET-UP-GEOMETRY=NOT
 SYMMETRIC
 No demons found for task LAB-SET-UP-GEOMETRY
 Constrained question posed to user for LAB-SET-UP GEOMETRY
 User responses = 3
 Apply constraints: LAB-SET-UP GEOMETRY = -ASYMMETRIC-EXIT
 Begin interrupt check for goal LAB-SET-UP-GEOMETRY=NOT
 ASYMMETRIC-EXIT
 No demons found for task LAB-SET-UP-GEOMETRY
 Apply constraints: LAB-SET-UP GEOMETRY = -ASYMMETRIC-
 GLANCING
 Begin interrupt check for goal LAB-SET-UP-GEOMETRY=NOT
 ASYMMETRIC-GLANCING
 No demons found for task LAB-SET-UP-GEOMETRY
 Apply constraints: LAB-SET-UP GEOMETRY = -ASYMMETRIC-SKEWED
 Begin interrupt check for goal LAB-SET-UP-GEOMETRY=NOT
 ASYMMETRIC-SKEWED
 No demons found for task LAB-SET-UP-GEOMETRY
 Value SYMMETRIC-SKEWED added to frame LAB-SET-UP for
 GEOMETRY task
 Begin interrupt check for goal LAB-SET-UP-
 GEOMETRY=SYMMETRIC-SKEWED
 No demons found for task LAB-SET-UP-GEOMETRY

Default question posed to user for LAB-SET-UP REFLECTION-INDICIES with default 004

User responses = Y

Apply constraint: LAB-SET-UP REFLECTION-INDICIES = ~044

Begin interrupt check for goal LAB-SET-UP-REFLECTION-INDICIES=NOT 044

No demons found for task LAB-SET-UP-REFLECTION-INDICIES

Apply constraint: LAB-SET-UP REFLECTION-INDICIES = ~113

Begin interrupt check for goal LAB-SET-UP-REFLECTION-INDICIES=NOT 113

No demons found for task LAB-SET-UP-REFLECTION-INDICIES

Apply constraint: LAB-SET-UP REFLECTION-INDICIES = ~224

Begin interrupt check for goal LAB-SET-UP-REFLECTION-INDICIES=NOT 224

No demons found for task LAB-SET-UP-REFLECTION-INDICIES

Apply constraint: LAB-SET-UP REFLECTION-INDICIES = ~115

Begin interrupt check for goal LAB-SET-UP-REFLECTION-INDICIES=NOT 115

No demons found for task LAB-SET-UP-REFLECTION-INDICIES

Value 004 added to frame LAB-SET-UP for REFLECTION-INDICIES task

Begin interrupt check for goal LAB-SET-UP-REFLECTION-INDICIES=004

No demons found for task LAB-SET-UP-REFLECTION-INDICIES

Default question posed to user for LAB-SET-UP STRUCTURE with default CUBIC

User responses = Y

Apply constraint: LAB-SET-UP STRUCTURE = ~TETRAGONAL

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=NOT TETRAGONAL

No demons found for task LAB-SET-UP-STRUCTURE

Apply constraint: LAB-SET-UP STRUCTURE = ~ORTHORHOMBIC

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=NOT ORTHORHOMBIC

No demons found for task LAB-SET-UP-STRUCTURE

Apply constraint: LAB-SET-UP STRUCTURE = -RHOMBOHEDRAL

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=NOT RHOMBOHEDRAL

No demons found for task LAB-SET-UP-STRUCTURE

Apply constraint: LAB-SET-UP STRUCTURE = -HEXAGONAL

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=NOT HEXAGONAL

No demons found for task LAB-SET-UP-STRUCTURE

Apply constraint: LAB-SET-UP STRUCTURE = -MONOCLINIC

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=NOT MONOCLINIC

No demons found for task LAB-SET-UP-STRUCTURE

Apply constraint: LAB-SET-UP STRUCTURE = -TRICLINIC

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=NOT TRICLINIC

No demons found for task LAB-SET-UP-STRUCTURE

Value CUBIC added to frame LAB-SET-UP for STRUCTURE task

Begin interrupt check for goal LAB-SET-UP-STRUCTURE=CUBIC

No demons found for task LAB-SET-UP-STRUCTURE

BEGIN BACKWARD CHAINING

Prove goal MILLER-INDICIES IS 001 from RULE1

Proof Tree from rules:

Prove goal MILLER-INDICIES IS 001 from RULE6

Proof Tree from rules:

Production Rule question posed to user for SKEW-ANGLE IS ZERO-DEGREES

User response = ZERO-DEGREES

Begin interrupt check for goal SKEW-ANGLE=ZERO-DEGREES

No demons found for task SKEW-ANGLE

Production Rule question posed to user for WAFER-MISORIENTATION IS ZERO

User response = ZERO

Begin interrupt check for goal WAFER-MISORIENTATION=ZERO

No demons found for task WAFER-MISORIENTATION

Prove goal MILLER-INDICIES IS 001 from RULE7

Proof Tree from rules:

Begin interrupt check for goal MILLER-INDICIES=001

No demons found for task MILLER-INDICIES

Value 001 added to frame LAB-SET-UP for MILLER-INDICIES task

Begin interrupt check for goal LAB-SET-UP-MILLER-INDICIES=001

No demons found for task LAB-SET-UP-MILLER-INDICIES

Constrained question posed to user for LAB-SET-UP ARC-RANGE
User responses = 100

Value 100 added to frame LAB-SET-UP for ARC-RANGE task

Begin interrupt check for goal LAB-SET-UP-ARC-RANGE=100

No demons found for task LAB-SET-UP-ARC-RANGE

Constrained question posed to user for LAB-SET-UP STEPS

User responses = .6

Value .6 added to frame LAB-SET-UP for STEPS task

Begin interrupt check for goal LAB-SET-UP-STEPS=.6

No demons found for task LAB-SET-UP-STEPS

Value NARROW added to frame LAB-SET-UP for SCAN task

Begin interrupt check for goal LAB-SET-UP-SCAN=NARROW

No demons found for task LAB-SET-UP-SCAN

BEGIN BACKWARD CHAINING

Prove goal REFERENCE-CRYSTAL IS UNNECESSARY from RULE2

Proof Tree from rules:

Prove goal REFERENCE-CRYSTAL IS NOT-YET-NECESSARY from RULE5

Proof Tree from rules:

Prove goal REFERENCE-CRYSTAL IS NOT SECOND-CRYSTAL from
RULE3"

Proof Tree from rules:

Production Rule question posed to user for MISMATCH IS HIGH

User response = HIGH

Begin interrupt check for goal MISMATCH=HIGH

No demons found for task MISMATCH

Production Rule question posed to user for LAB-SET-UP-SCAN
HAS WIDE

User response = WIDE

Begin interrupt check for goal LAB-SET-UP-SCAN=WIDE

No demons found for task LAB-SET-UP-SCAN

Production Rule question posed to user for LAB-SET-UP-SCAN
HAS VERY-WIDE

User response = VERY-WIDE

Begin interrupt check for goal LAB-SET-UP-SCAN=VERY-WIDE

No demons found for task LAB-SET-UP-SCAN

Production Rule question posed to user for FEATURE-TYPE HAS
COMPLEX-ROCKING-CURVE

User response = COMPLEX-ROCKING-CURVE

Begin interrupt check for goal FEATURE-TYPE=COMPLEX-ROCKING-
CURVE

No demons found for task FEATURE-TYPE

Production Rule question posed to user for SIMULATION IS
RECOMMENDED

Negated user response = -RECOMMENDED

Begin interrupt check for goal SIMULATION=NOT RECOMMENDED

Interrupt DEMON1 fails for task SIMULATION

FIND NEW TASKS FOR AGENDA

COMPOSITION GEOMETRY REFLECTION-INDICIES tasks found for
TARGET analog

COMPOSITION task placed on agenda with priority of .375
GEOMETRY task placed on agenda with priority of .35
REFLECTION-INDICIES task placed on agenda with priority of .325

BEGIN FORWARD CHAINING ON TARGET

Constrained question posed to user for REFERENCE-CRYSTAL COMPOSITION

User responses = 1

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -Ge

Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT Ge

No demons found for task REFERENCE-CRYSTAL-COMPOSITION

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -GaAs

Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT GaAs

No demons found for task REFERENCE-CRYSTAL-COMPOSITION

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -InAs

Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT InAs

No demons found for task REFERENCE-CRYSTAL-COMPOSITION

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -AlAs

Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT AlAs

No demons found for task REFERENCE-CRYSTAL-COMPOSITION

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -AlSb

Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT AlSb

No demons found for task REFERENCE-CRYSTAL-COMPOSITION

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -GaSb

Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT GaSb

No demons found for task REFERENCE-CRYSTAL-COMPOSITION

Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -InSb
Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT InSb
No demons found for task REFERENCE-CRYSTAL-COMPOSITION
Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -AlP
Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT AlP
No demons found for task REFERENCE-CRYSTAL-COMPOSITION
Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -GaP
Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT GaP
No demons found for task REFERENCE-CRYSTAL-COMPOSITION
Apply constraints: REFERENCE-CRYSTAL COMPOSITION = -InP
Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=NOT InP
No demons found for task REFERENCE-CRYSTAL-COMPOSITION
Value Si added to frame REFERENCE-CRYSTAL for COMPOSITION task
Begin interrupt check for goal REFERENCE-CRYSTAL-COMPOSITION=Si
No demons found for task REFERENCE-CRYSTAL-COMPOSITION
Default question posed to user for REFERENCE-CRYSTAL GEOMETRY with default SYMMETRIC
User responses = Y
Apply constraint: REFERENCE-CRYSTAL GEOMETRY = -SYMMETRIC
Begin interrupt check for goal REFERENCE-CRYSTAL-GEOMETRY=NOT SYMMETRIC
No demons found for task REFERENCE-CRYSTAL-GEOMETRY
Apply constraint: REFERENCE-CRYSTAL GEOMETRY = -GLANCING-INCIDENT
Begin interrupt check for goal REFERENCE-CRYSTAL-GEOMETRY=NOT GLANCING-INCIDENT
No demons found for task REFERENCE-CRYSTAL-GEOMETRY

Apply constraint: REFERENCE-CRYSTAL GEOMETRY = -GLANCING-EXIT

Begin interrupt check for goal REFERENCE-CRYSTAL-GEOMETRY=NOT GLANCING-EXIT

No demons found for task REFERENCE-CRYSTAL-GEOMETRY

Value SYMMETRIC added to frame REFERENCE-CRYSTAL for GEOMETRY task

Default question posed to user for REFERENCE-CRYSTAL-REFLECTION-INDICIES with default 004

User responses = Y

Apply constraint: REFERENCE-CRYSTAL REFLECTION-INDICIES = -044

Begin interrupt check for goal REFERENCE-CRYSTAL-REFLECTION-INDICIES=NOT 044

No demons found for task REFERENCE-CRYSTAL-REFLECTION-INDICIES

Apply constraint: REFERENCE-CRYSTAL REFLECTION-INDICIES = -113

Begin interrupt check for goal REFERENCE-CRYSTAL-REFLECTION-INDICIES=NOT 113

No demons found for task REFERENCE-CRYSTAL-REFLECTION-INDICIES

Apply constraint: REFERENCE-CRYSTAL REFLECTION-INDICIES = -224

Begin interrupt check for goal REFERENCE-CRYSTAL-REFLECTION-INDICIES=NOT 224

No demons found for task REFERENCE-CRYSTAL-REFLECTION-INDICIES

Apply constraint: REFERENCE-CRYSTAL REFLECTION-INDICIES = -115

Begin interrupt check for goal REFERENCE-CRYSTAL-REFLECTION-INDICIES=NOT 115

No demons found for task REFERENCE-CRYSTAL-REFLECTION-INDICIES

Value 004 added to frame REFERENCE-CRYSTAL for REFLECTION-INDICIES task

Begin interrupt check for goal REFERENCE-CRYSTAL-REFLECTION-INDICIES=004

No demons found for task REFERENCE-CRYSTAL-REFLECTION-INDICIES

FIND NEW TASKS FOR AGENDA

No tasks found for TARGET analog

FIND NEW TASKS FOR AGENDA

No tasks found for SOURCE analog

FIND NEW TASKS FOR AGENDA

No tasks found for MAP analog

FIND NEW TASKS FOR AGENDA

No tasks found for VALUATE analog

FIND NEW TASKS FOR AGENDA

Value Si added to frame LAB-SET-UP for REFERENCE-CRYSTAL task

Begin interrupt check for goal LAB-SET-UP-REFERENCE-CRYSTAL=Si

No demons found for task LAB-SET-UP-REFERENCE-CRYSTAL

Value SYMMETRIC added to frame LAB-SET-UP for REFERENCE-CRYSTAL task

Begin interrupt check for goal LAB-SET-UP-REFERENCE-CRYSTAL=SYMMETRIC

No demons found for task LAB-SET-UP-REFERENCE-CRYSTAL

Value 004 added to frame LAB-SET-UP for REFERENCE-CRYSTAL task

Begin interrupt check for goal LAB-SET-UP-REFERENCE-CRYSTAL=004

No demons found for task LAB-SET-UP-REFERENCE-CRYSTAL

FIND NEW TASKS FOR AGENDA

No tasks found for TARGET analog

FIND NEW TASKS FOR AGENDA

No tasks found for SOURCE analog

FIND NEW TASKS FOR AGENDA

No tasks found for MAP analog

FIND NEW TASKS FOR AGENDA

No tasks found for VALUATE analog

FIND NEW TASKS FOR AGENDA

Value Cu added to frame ALL-ROCKING-CURVES for LAB-SET-UP
task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-
UP=Cu

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value SYMMETRIC-SKEWED added to frame ALL-ROCKING-CURVES for
LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-
UP=SYMMETRIC-SKEWED

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value 004 added to frame ALL-ROCKING-CURVES for LAB-SET-UP
task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-
UP=004

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value CUBIC added to frame ALL-ROCKING-CURVES for LAB-SET-UP
task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-
UP=CUBIC

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value 001 added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-UP=001

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value Si added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-UP=Si

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value SYMMETRIC added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-UP=SYMMETRIC

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value 004 added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Value 100 added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-UP=100

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value .6 added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-UP=.6

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Value NARROW added to frame ALL-ROCKING-CURVES for LAB-SET-UP task

Begin interrupt check for goal ALL-ROCKING-CURVES-LAB-SET-UP=NARROW

No demons found for task ALL-ROCKING-CURVES-LAB-SET-UP

Probability of cause for TARGET analog

Old probability $P(H E)_o$	= .4
Base probability $P(H)$	= .25
Probability of new evidence $P(E)$	= .933333
Probability given new hypothesis $P(H E)_n$	= .756757

Probability differential with P|E adjustment = .178378
Adjusted probability P|E.P(H|E)n = .57

Target analog = .578378
Source analog = .210811
Map analog = .140541
Evaluate analog = 7.02703E-2

MQW frame uninstantiated

FIND NEW TASKS FOR AGENDA

FRINGES NOISE THIN-LAYER tasks found for TARGET analog.

FRINGES task placed on agenda with priority of .486486
NOISE task placed on agenda with priority of .394595
THIN-LAYER task placed on agenda with priority of .302703

BEGIN FORWARD CHAINING ON TARGET

BEGIN BACKWARD CHAINING

BEGIN FORWARD CHAINING ON TARGET

BEGIN BACKWARD CHAINING

BEGIN FORWARD CHAINING ON TARGET

Probability of cause for TARGET analog

Old probability P(H E)o	= .578378
Base probability P(H)	= .25
Probability of new evidence P(E)	= 0.0
Probability given new hypothesis P(H E)n	= 0.0
Probability differential with P E adjustment	= -5.78378E-2
Adjusted probability P E.P(H E)n	= .52054

Target analog = .52054
Source analog = .23973
Map analog = .15982
Evaluate analog = 7.99099E-2

MQW frame uninstantiated

FIND NEW TASKS FOR AGENDA

No tasks found for TARGET analog

FIND NEW TASKS FOR AGENDA

FEATURE-TYPE PEAK-COUNT PEAK-DENSITY QUALITY tasks found
for SOURCE analog.

FEATURE-TYPE task placed on agenda with priority of .22374

PEAK-COUNT task placed on agenda with priority of .207766

PEAK-DENSITY task placed on agenda with priority of .191784

QUALITY task placed on agenda with priority of .175802

BEGIN FORWARD CHAINING ON SOURCE

BEGIN BACKWARD CHAINING

Value COMPLEX-ROCKING-CURVE added to frame MQW for FEATURE-
TYPE task

Begin interrupt check for goal MQW-FEATURE-TYPE=COMPLEX-
ROCKING-CURVE

No demons found for task MQW-FEATURE-TYPE

Update probability matrix for FEATURE-TYPE task

BEGIN BACKWARD CHAINING

BEGIN BACKWARD CHAINING

BEGIN BACKWARD CHAINING

Probability of cause for SOURCE analog

Old probability $P(H E)_o$	= .23973
Base probability $P(H)$	= .25
Probability of new evidence $P(E)$	= .25
Probability given new hypothesis $P(H E)_n$	= 3.38498E-2
Probability differential with P E adjustment	= -8.23519E-2
Adjusted probability $P E.P(H E)_n$	= .157378

Target analog = .576925
Source analog = .157378
Map analog = .177131
Evaluate analog = 8.85657E-2

MQW frame uninstantiated

FIND NEW TASKS FOR AGENDA

No tasks found for TARGET analog

FIND NEW TASKS FOR AGENDA

Searching probability matrix for SOURCE data to MAP to MQW

Start location is: 10 10 10

Map to target MQW from source EX101010

No tasks found for MAP analog

FIND NEW TASKS FOR AGENDA

No tasks found for SOURCE analog

FIND NEW TASKS FOR AGENDA

No tasks found for VALUATE analog

FIND NEW TASKS FOR AGENDA

MQW frame uninstantiated

CONSULTATION ENDS

SET DEFINITIONS FOR EVALUATION

Target set = FEATURE-TYPE PEAK-COUNT PEAK-DENSITY
QUALITY Target sub-set = FEATURE-TYPE
Source set =
Source sub-set =
Universal set = FEATURE-TYPE PEAK-COUNT PEAK-DENSITY
QUALITY

PROBABILITY VALUES FOR TASKS IN TARGET AND SOURCE SETS

Intersect of tasks with intersect values	= 0.0
Intersect of tasks with any value	= 0.0
Union of all tasks with no values	= .75
Intersect of tasks with or without values	= 0.0

EVALUATION OF SET PROBABILITIES

Probability of matched task	= 0.0
Confidence in matching process	= .25
Similarity between TARGET and SOURCE	= 0.0
Closeness of analogy between TARGET and SOURCE	= 8.33333E-2

SET DEFINITIONS FOR EVALUATION

Target set = FEATURE-TYPE PEAK-COUNT PEAK-DENSITY
 QUALITY Target sub-set = FEATURE-TYPE
 Source set =
 Source sub-set =
 Universal set = FEATURE-TYPE PEAK-COUNT PEAK-DENSITY
 QUALITY

PROBABILITY VALUES FOR TASKS IN TARGET AND SOURCE SETS

Intersect of tasks with intersect values	= 0.0
Intersect of tasks with any value	= 0.0
Union of all tasks with no values	= .75
Intersect of tasks with or without values	= 0.0

EVALUATION OF SET PROBABILITIES

Probability of matched task	= 0.0
Confidence in matching process	= .25
Similarity between TARGET and SOURCE	= 0.0
Closeness of analogy between TARGET and SOURCE	= 8.33333E-2

SET DEFINITIONS FOR EVALUATION

Target set = FEATURE-TYPE PEAK-COUNT PEAK-DENSITY
 QUALITY Target sub-set = FEATURE-TYPE
 Source set =
 Source sub-set =
 Universal set = FEATURE-TYPE PEAK-COUNT PEAK-DENSITY
 QUALITY

PROBABILITY VALUES FOR TASKS IN TARGET AND SOURCE SETS

Intersect of tasks with intersect values	= 0.0
Intersect of tasks with any value	= 0.0
Union of all tasks with no values	= .75
Intersect of tasks with or without values	= 0.0

EVALUATION OF SET PROBABILITIES

Probability of matched task	= 0.0
Confidence in matching process	= .25
Similarity between TARGET and SOURCE	= 0.0
Closeness of analogy between TARGET and SOURCE	= 8.33333E-2

Appendix 5

Key of Abbreviations with Subject Area Underlined

Artificial Intelligence

A.I. = Artificial Intelligence
A* = special search algorithm
G-T = Generate and Test system

Cognitive Psychology

LTM = Long Term Memory
STM = Short Term Memory

Expert Systems

B.C. = Backward Chaining
E.S. = Expert System
F.C. = Forward Chaining
K.B. = Knowledge Base
K.S. = Knowledge Set

Knowledge Engineering

MDS = Multi Dimensional Scaling

Logic

DN = Double Negation
MPP = Modus Ponendo Ponens
MTT = Modus Tollendo Tollens
Wffs = Well formed formulae

Material Science

MQW = Multi-Quantum Well

Programming

CLOS = Common Lisp Object System

Statistical Analysis

ANOVA = ANalysis Of VAriance
SD = Standard Deviation

Statistical Reasoning

CF = Certainty Factors
NF = Necessity Factor.
SF = Sufficiency Factor